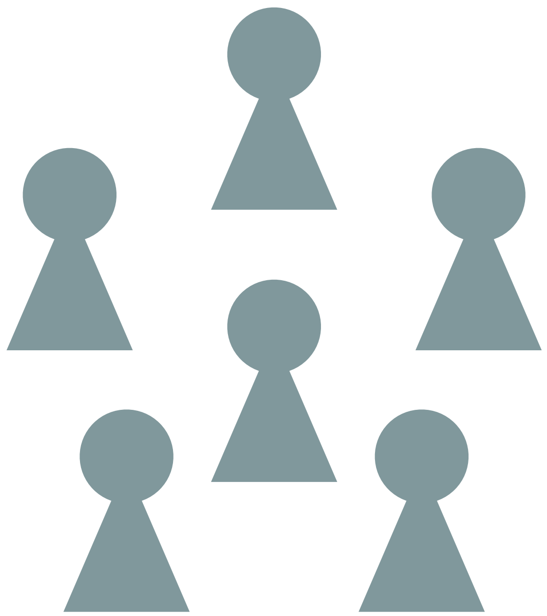
ANONYMOUS COMMUNICATION

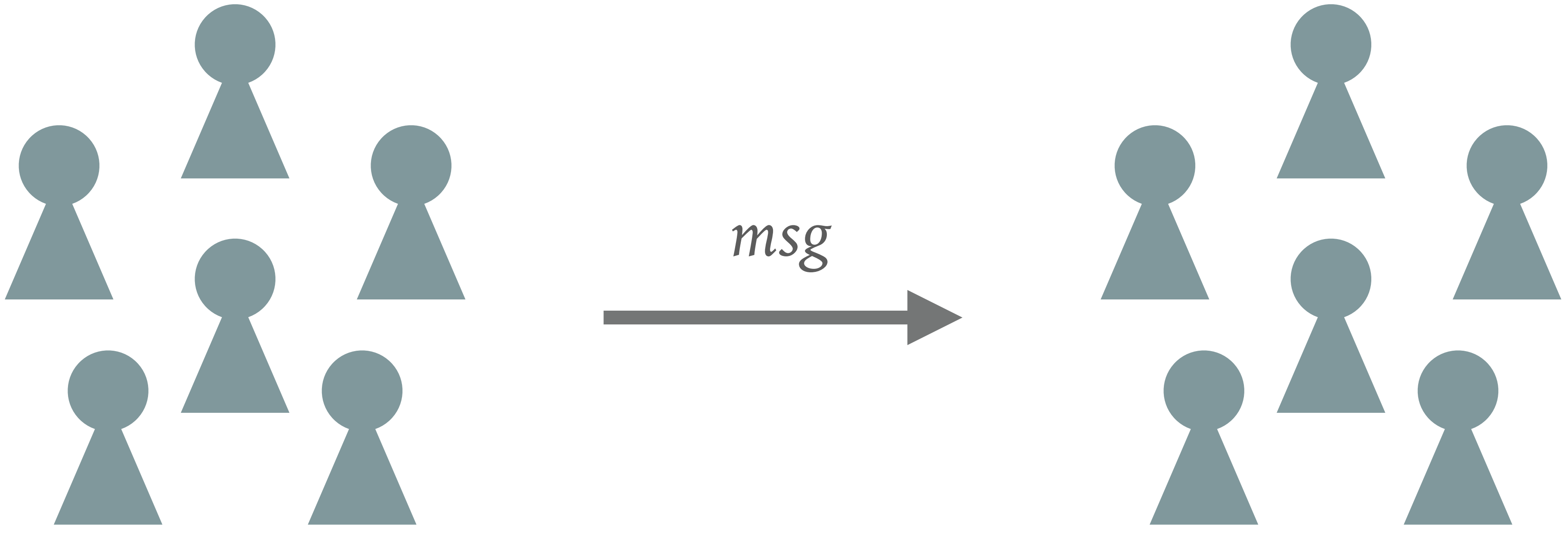
CMSC 414

APR 4 2019

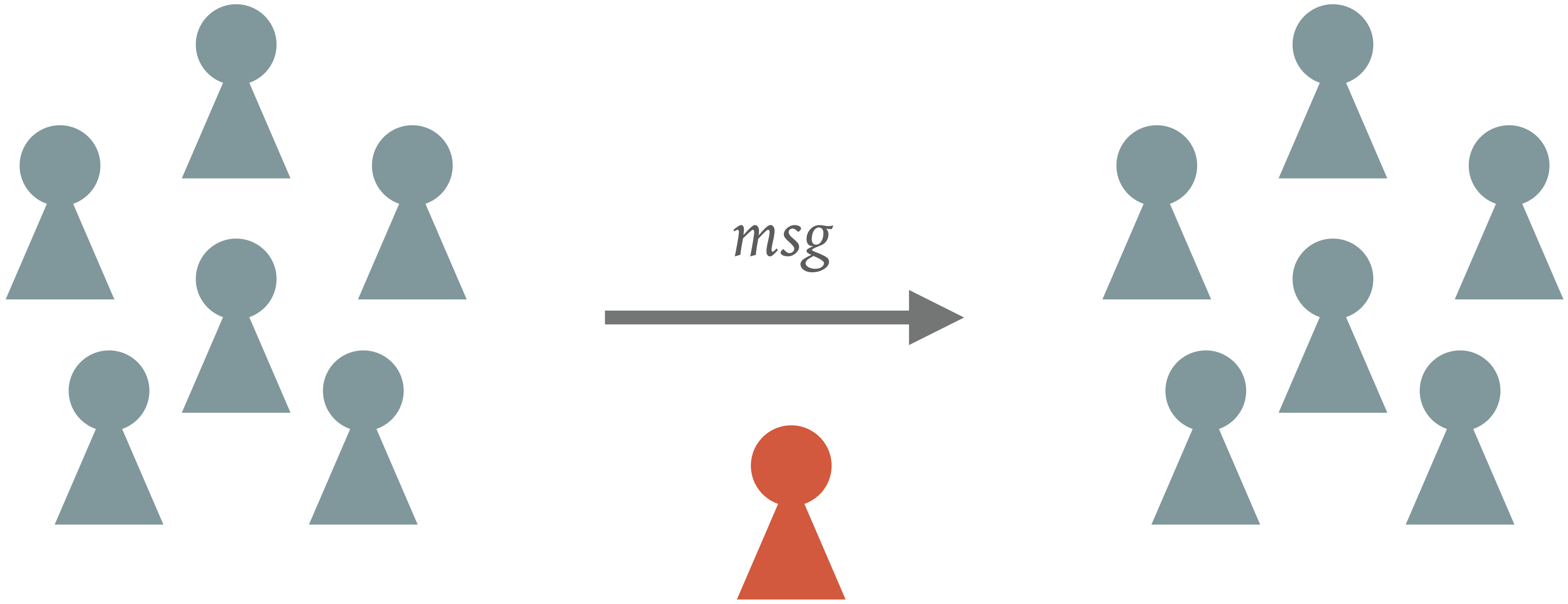
WHAT IS ANONYMITY?



WHAT IS ANONYMITY?

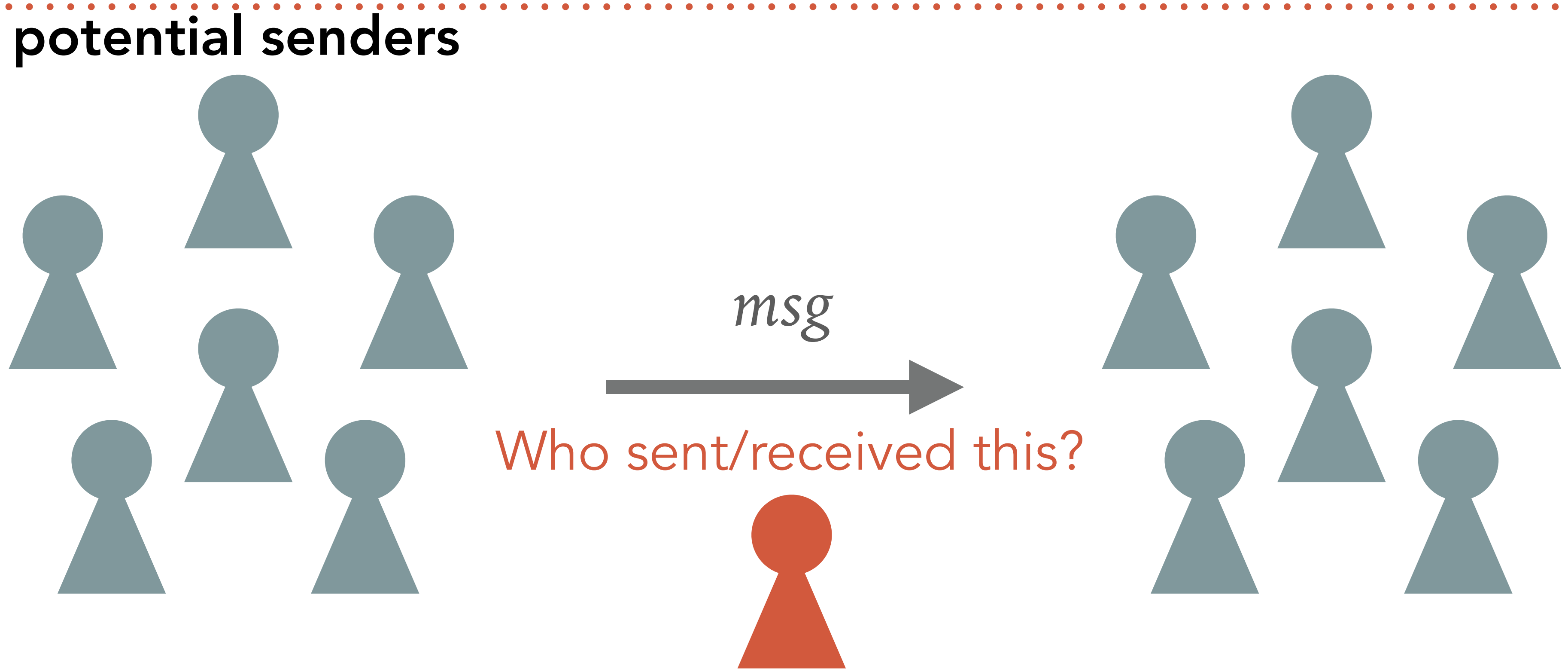


WHAT IS ANONYMITY?

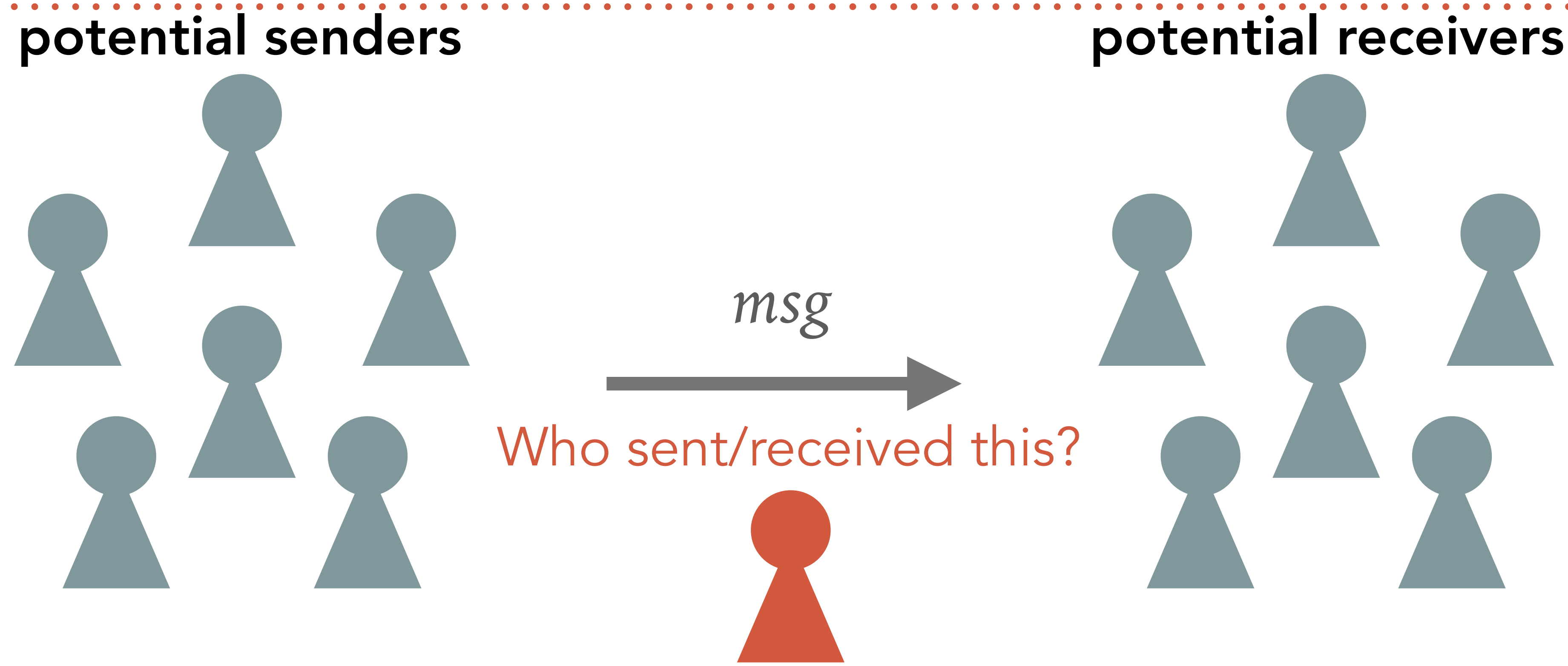


WHAT IS ANONYMITY?

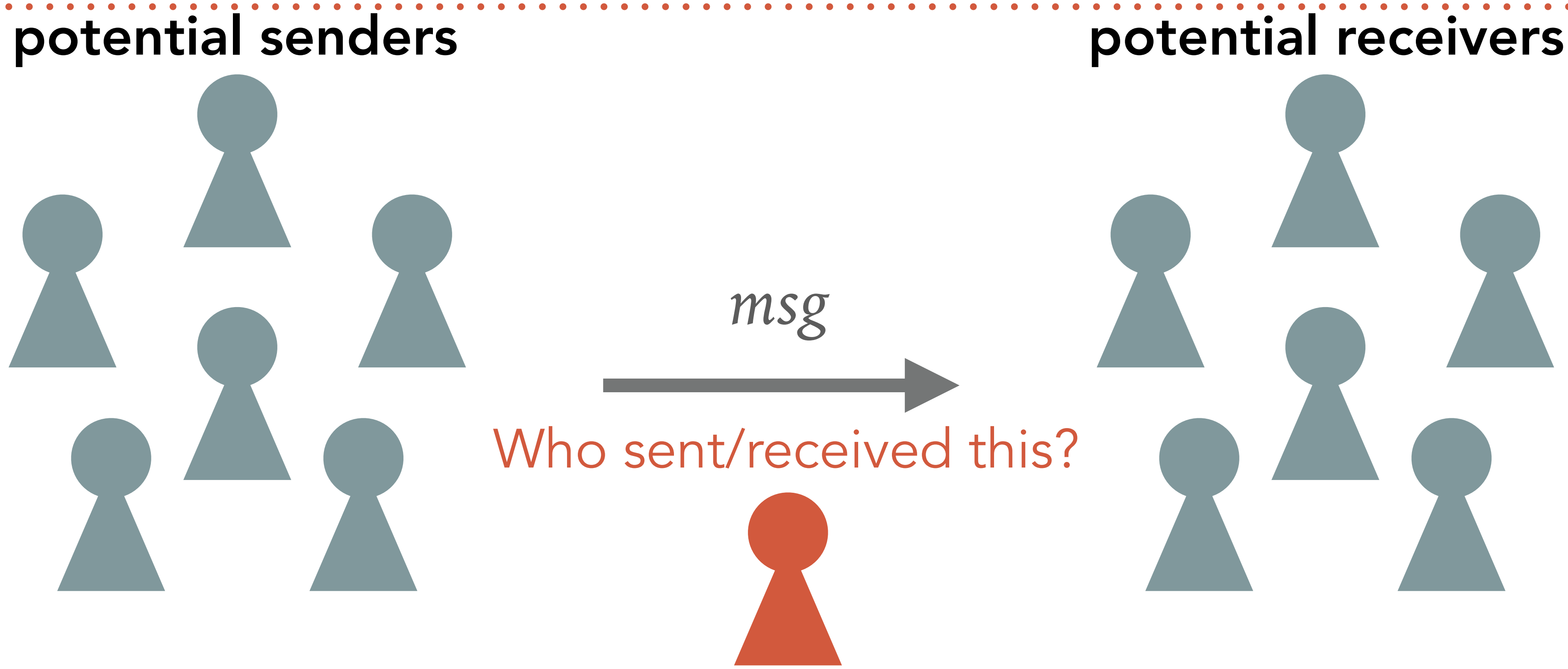
WHAT IS ANONYMITY?



WHAT IS ANONYMITY?



WHAT IS ANONYMITY?

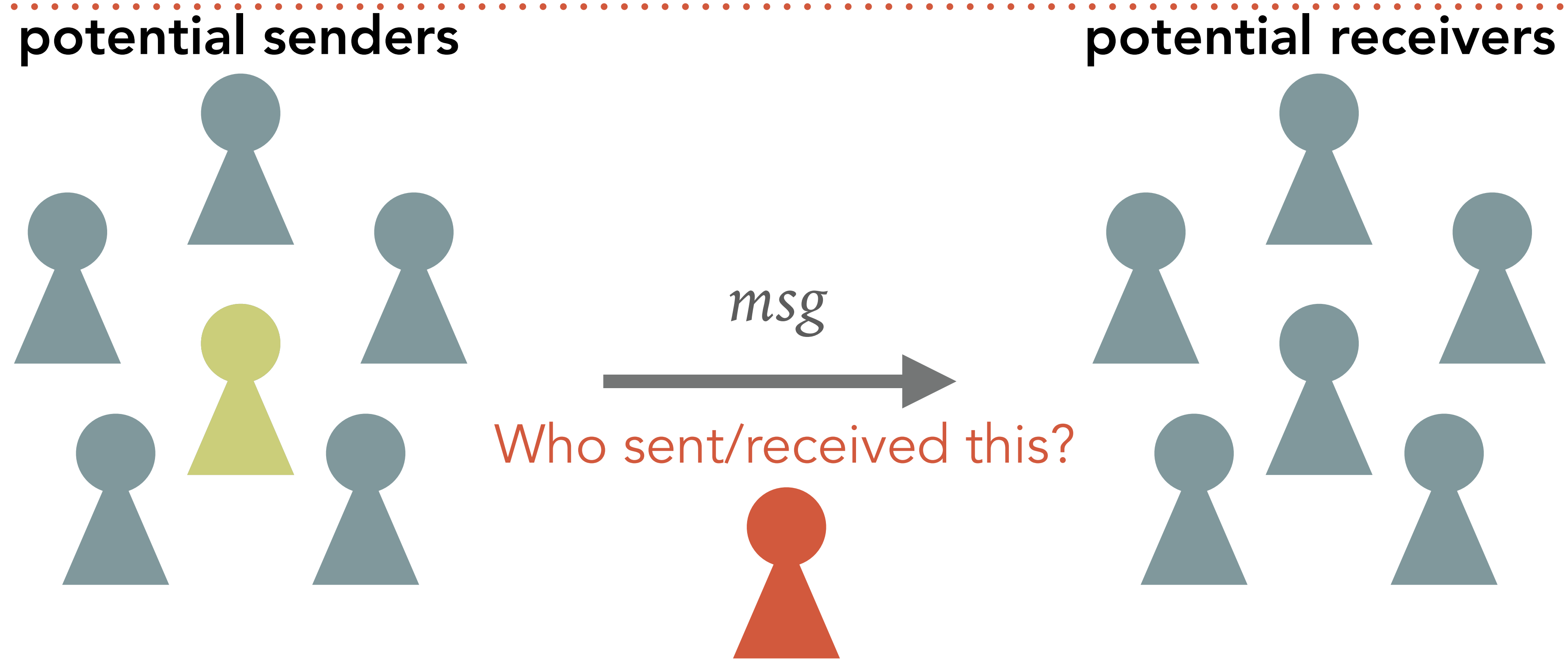


SENDER-ANONYMITY:

An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

(The attacker might learn the receiver)

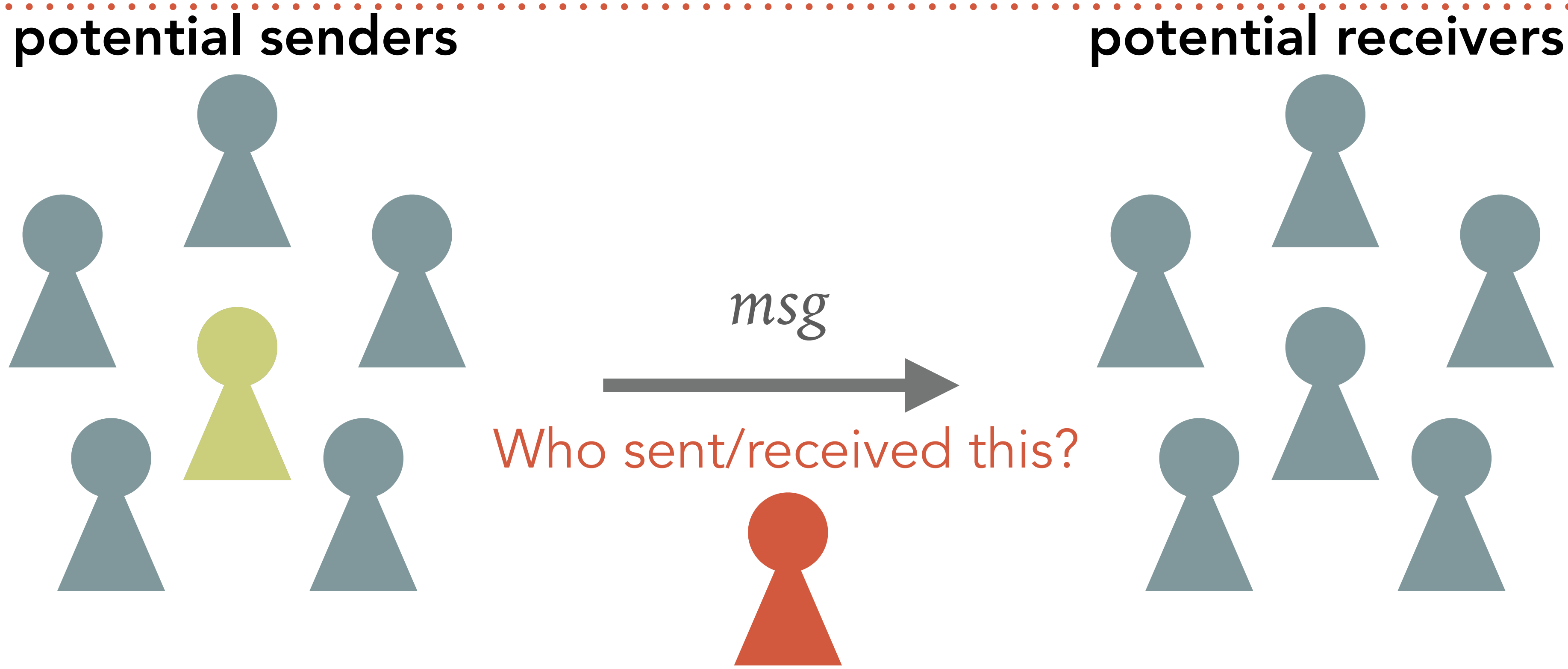
WHAT IS ANONYMITY?



SENDER-ANONYMITY:
An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

(The attacker might learn the receiver)

WHAT IS ANONYMITY?



SENDER-ANONYMITY:

An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

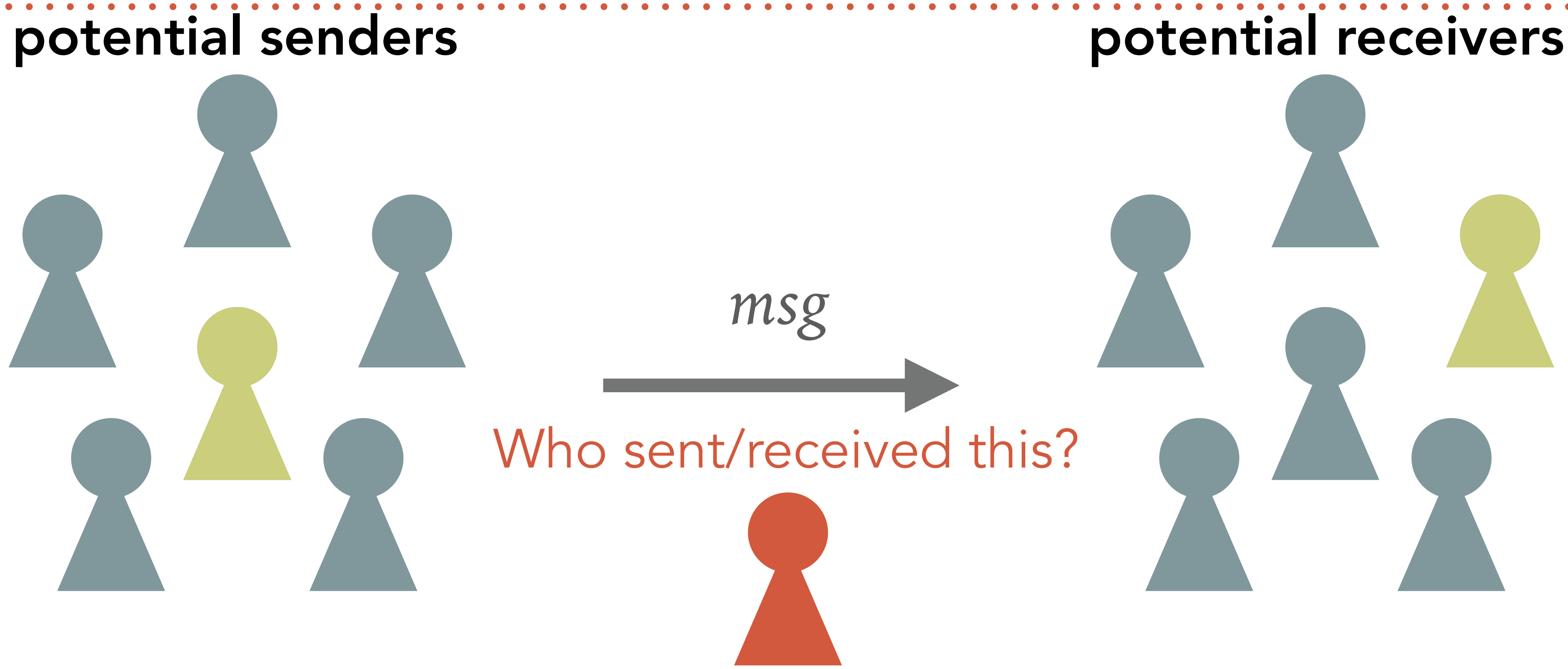
(The attacker might learn the receiver)

RECEIVER-ANONYMITY:

An attacker overhearing communication cannot determine the **true receiver** from a larger set of **potential receivers**.

(The attacker might learn the sender)

WHAT IS ANONYMITY?



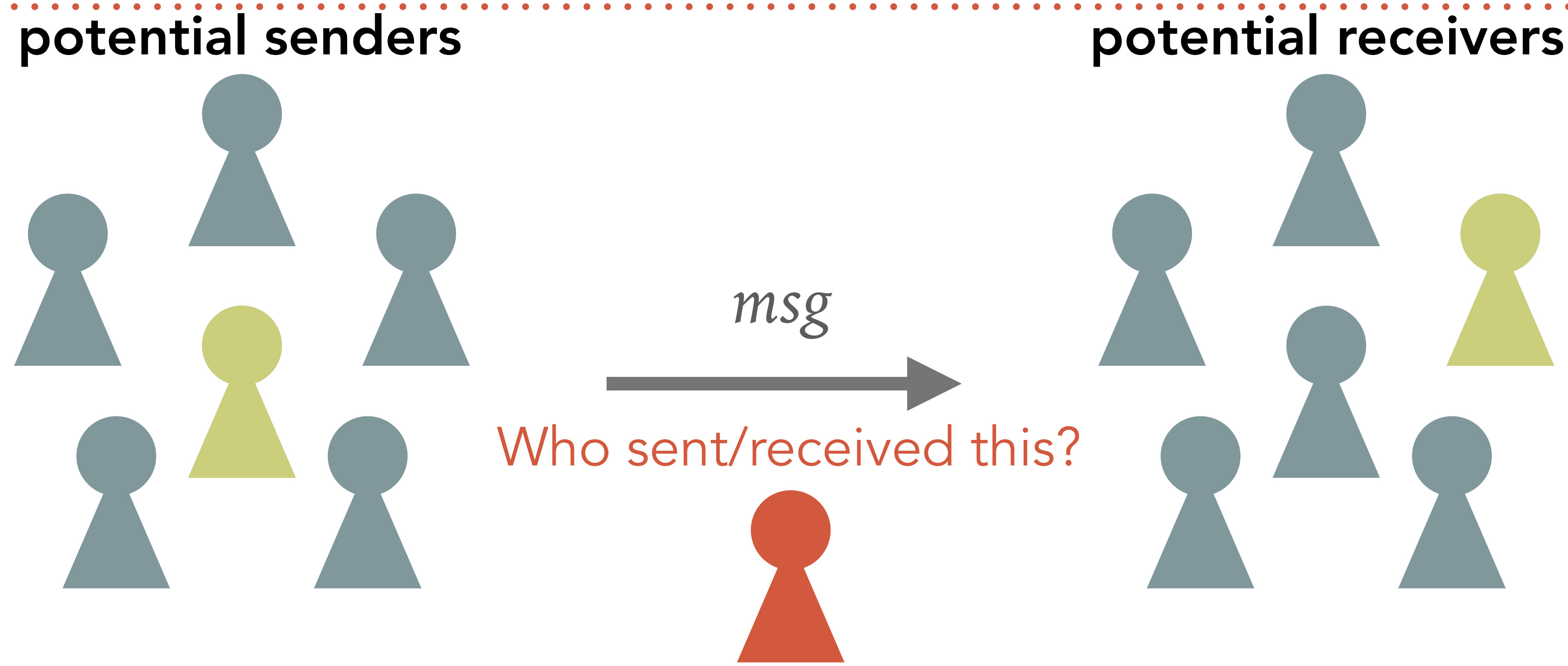
SENDER-ANONYMITY:
An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

(The attacker might learn the receiver)

RECEIVER-ANONYMITY:
An attacker overhearing communication cannot determine the **true receiver** from a larger set of **potential receivers**.

(The attacker might learn the sender)

WHAT IS ANONYMITY?



SENDER-ANONYMITY:
An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

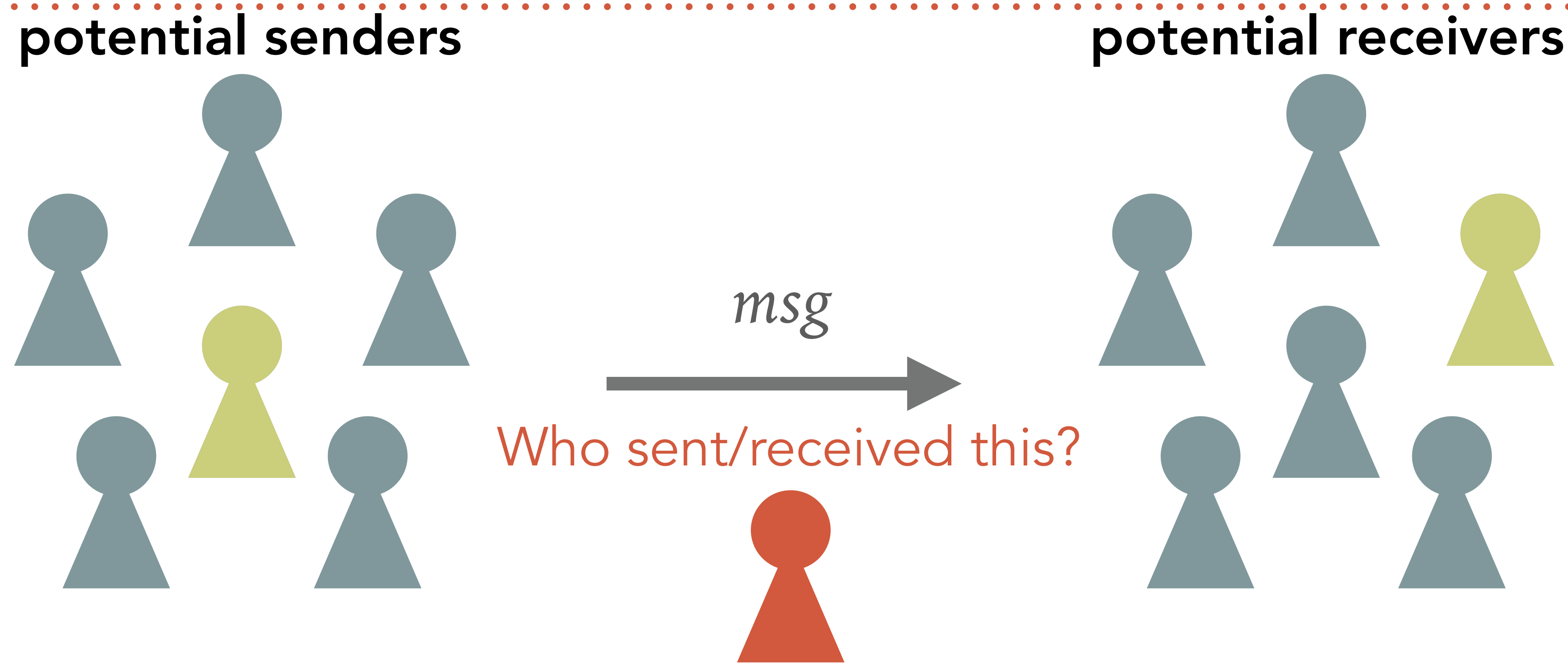
(The attacker might learn the receiver)

RECEIVER-ANONYMITY:
An attacker overhearing communication cannot determine the **true receiver** from a larger set of **potential receivers**.

(The attacker might learn the sender)

Number stations: Spies use small FM transmitters to broadcast encoded messages; the set of potential receivers is *everyone* within broadcast range

WHAT IS ANONYMITY?



SENDER-ANONYMITY:
An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

(The attacker might learn the receiver)

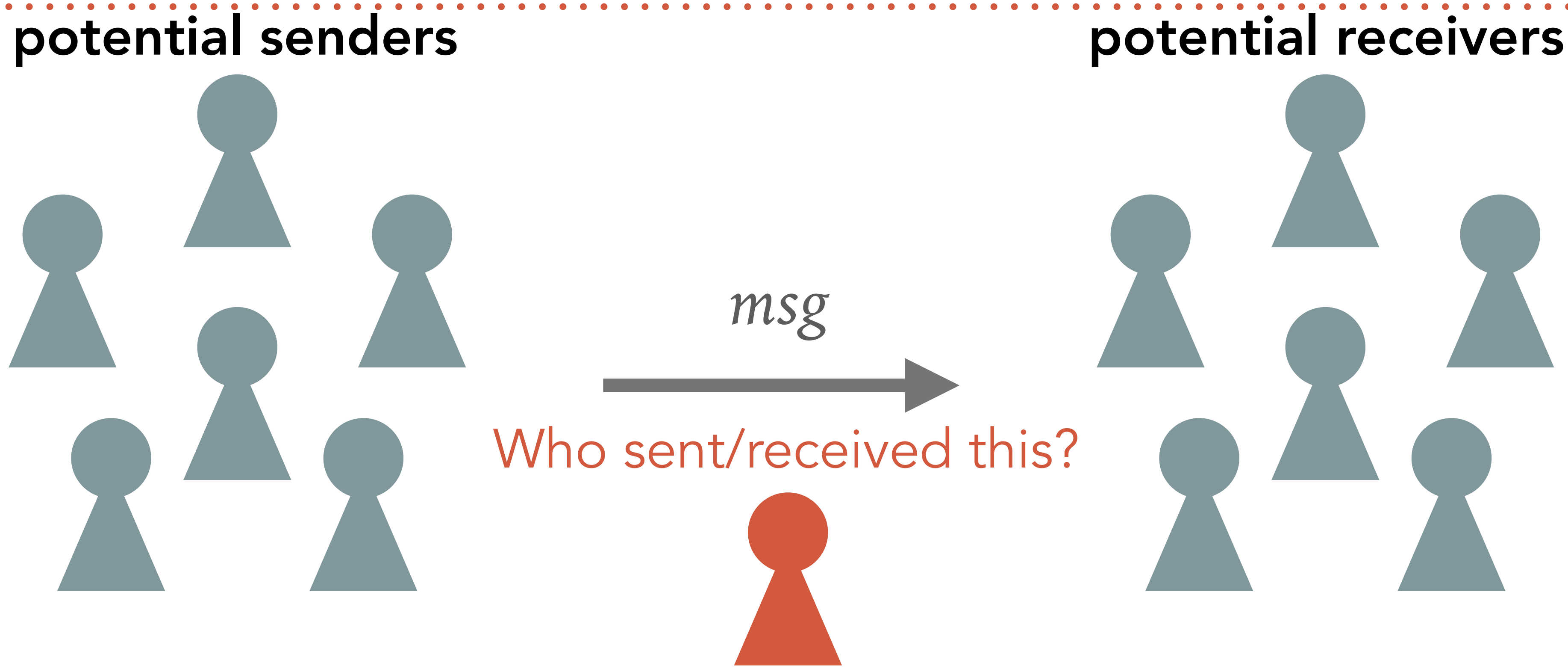
RECEIVER-ANONYMITY:
An attacker overhearing communication cannot determine the **true receiver** from a larger set of **potential receivers**.

(The attacker might learn the sender)

SENDER-RECEIVER-ANONYMITY:
An attacker overhearing communication cannot determine the **communicating pair** from a larger set of **potential pairs**.

(The attacker might learn the sender *or* the receiver, but not both)

QUANTIFYING ANONYMITY



ANONYMITY SET:
To quantify "how anonymous" a system / protocol is,
we think of how large the **anonymity set** is: the set of other potential users / computers that could have performed the action

Intuition:
The more other people it *might have been*, the less likely they can pin it to any individual user

Example:
In a densely populated area, the anonymity set of a number station can be tens of millions

ANONYMITY IS NOT PRIVACY (NOT EXACTLY)

Both of these are fungible terms, but generally speaking…

PRIVACY:
Maintaining confidentiality about an entity's **personally-identifying information (PII)**

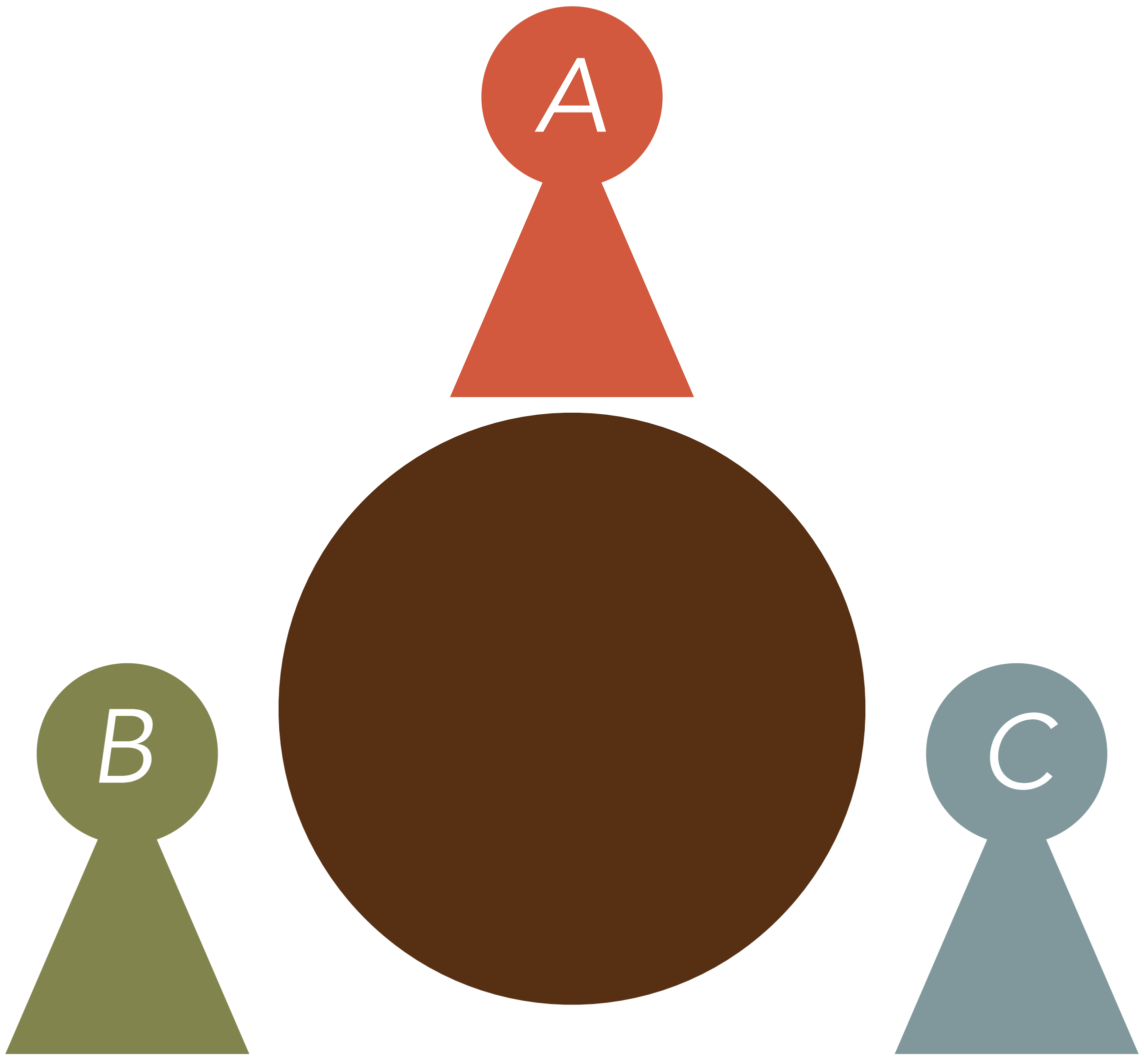
ANONYMITY:
Maintaining confidentiality about with whom (or whether) an entity communicates

The connection is complicated:
With whom you communicate is a form of PII

Sharing PII can de-anonymize your communication

This lecture: we will focus on anonymous communication

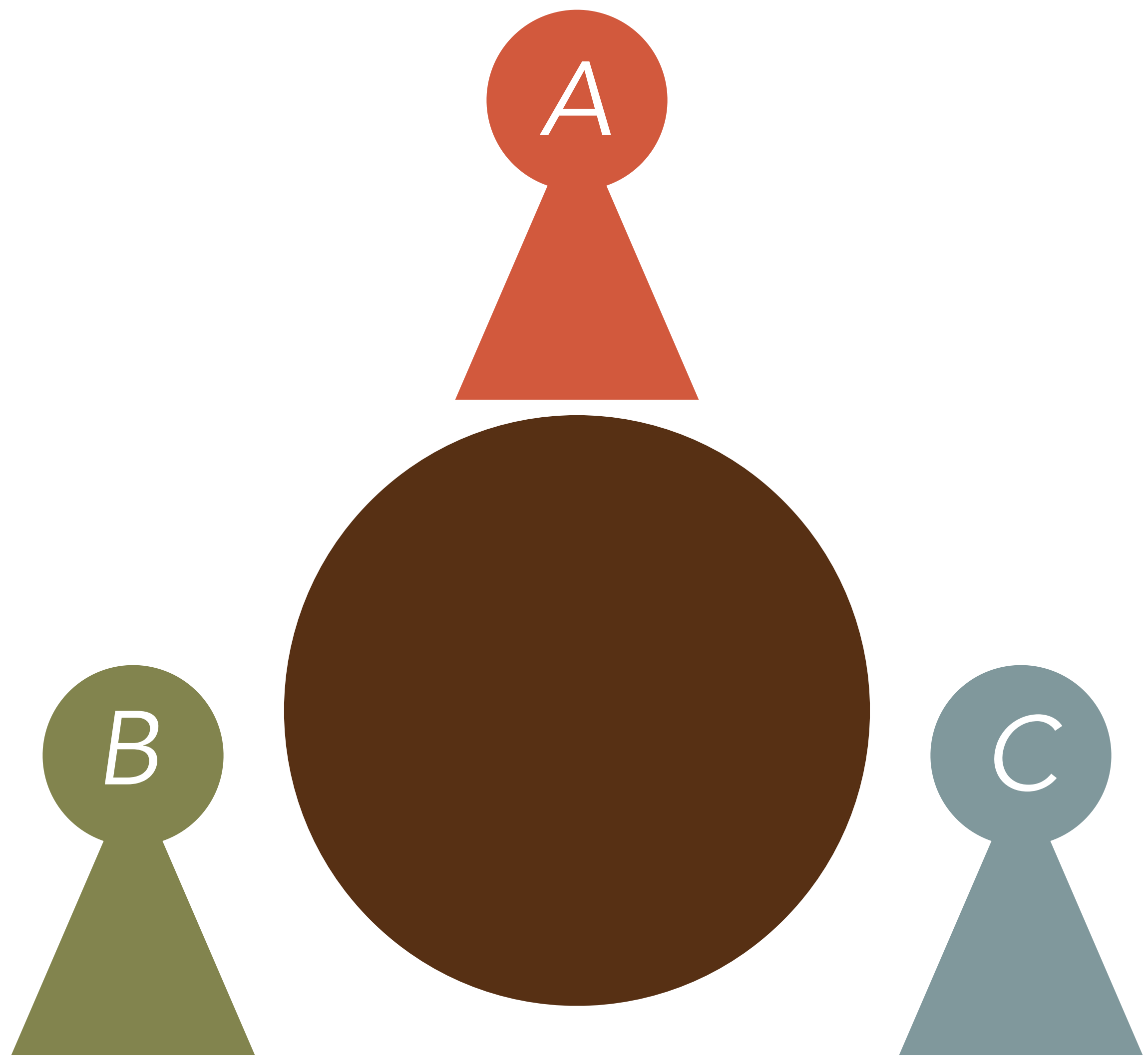
THE DINING CRYPTOGRAPHER'S PROBLEM



Each individual knows 2 bits

b_{left} and b_{right}

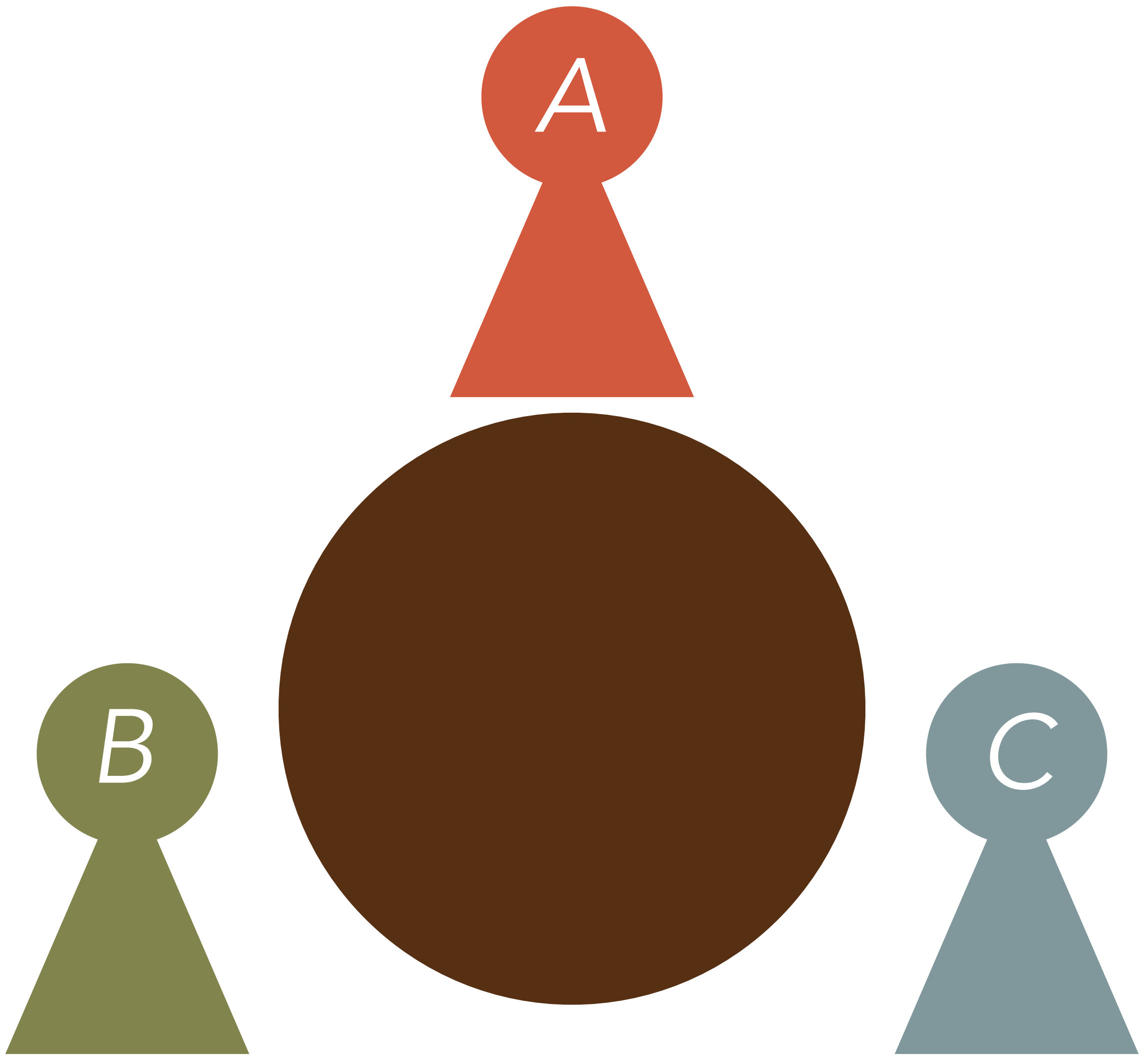
THE DINING CRYPTOGRAPHER'S PROBLEM



Each individual knows 2 bits
b_{left} and b_{right}

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM

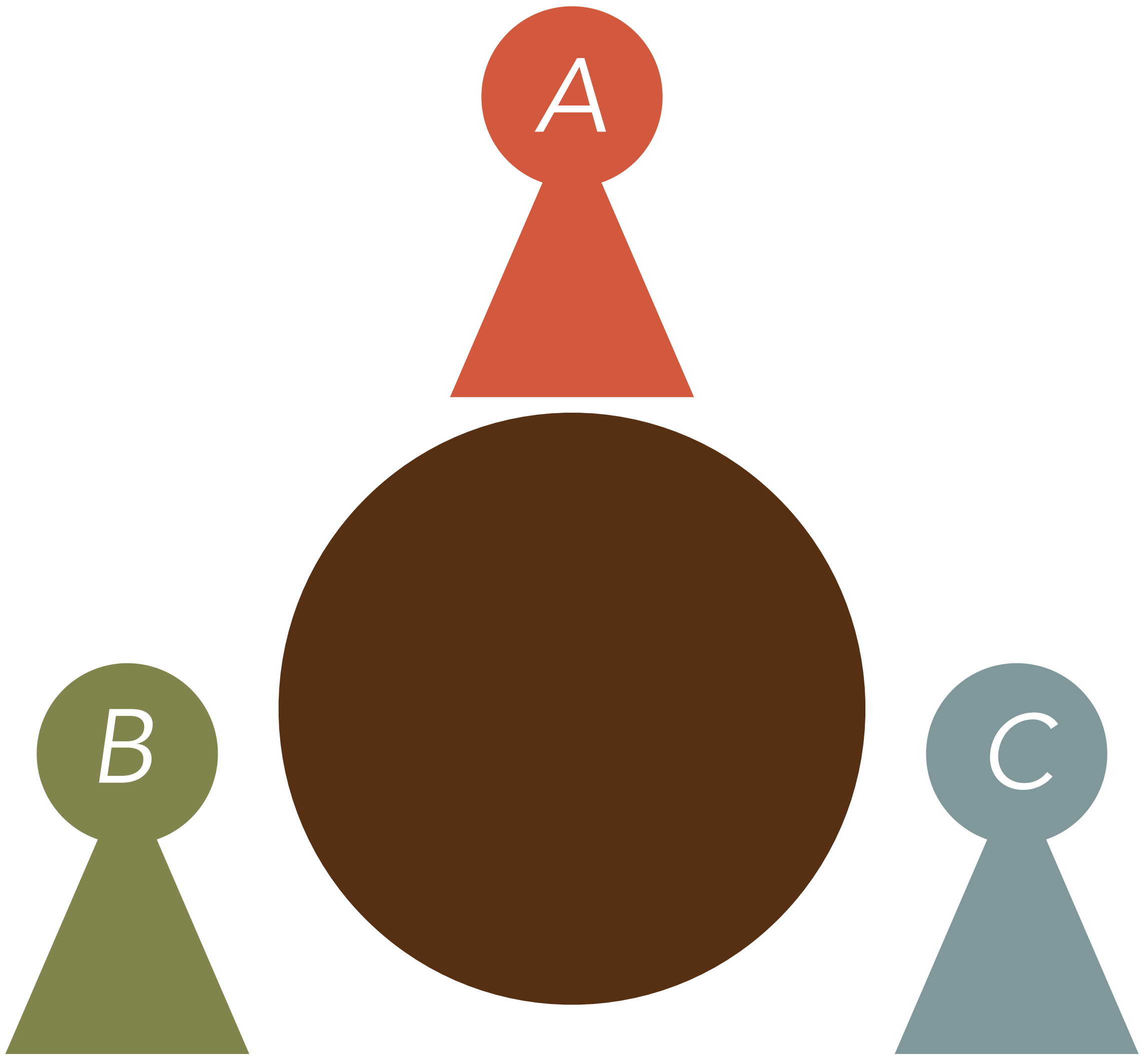


PROTOCOL

Each individual knows 2 bits
b_{left} and b_{right}

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



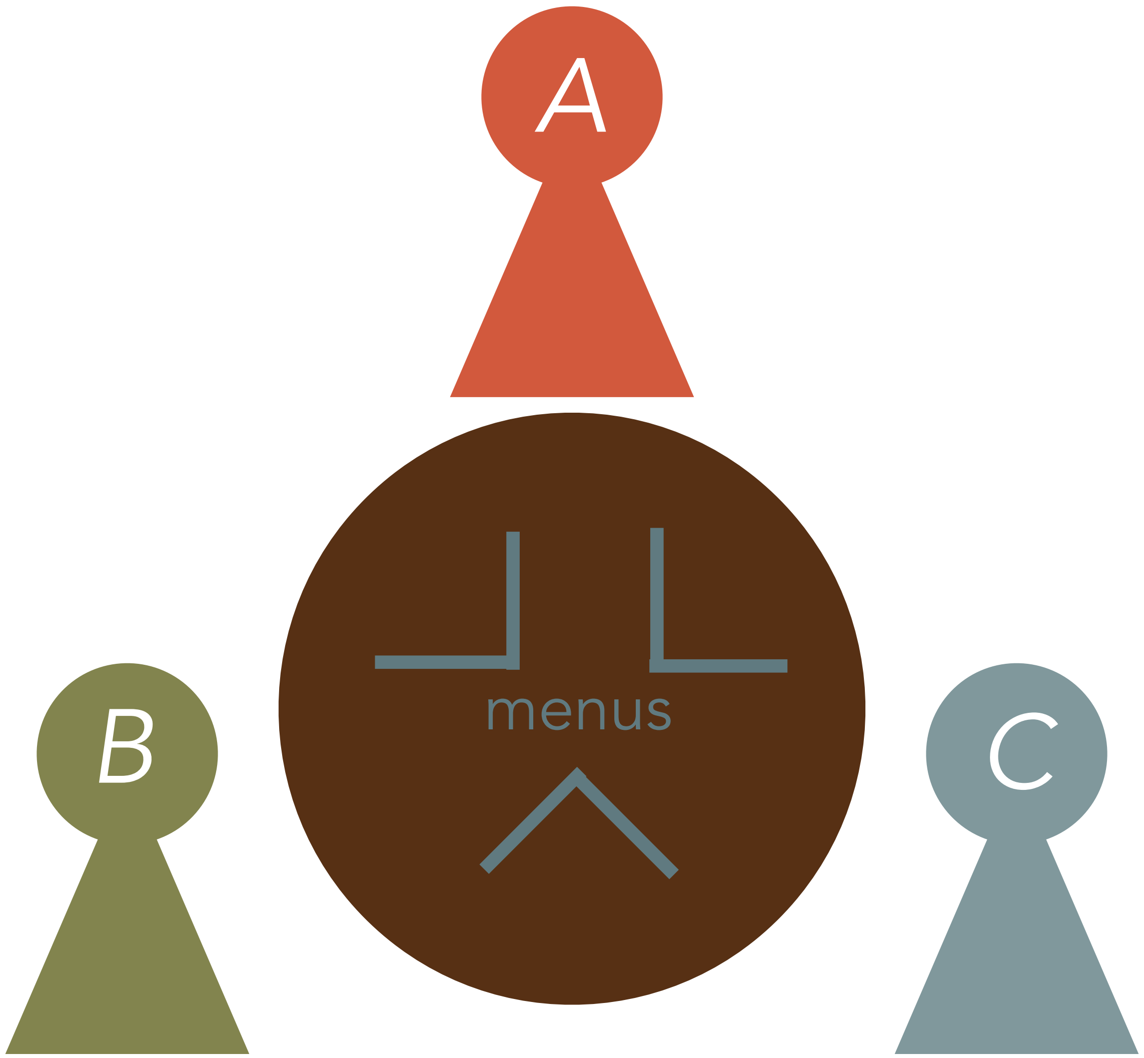
PROTOCOL

Each pair props up a menu
Private communication channel

Each individual knows 2 bits
b_{left} and b_{right}

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu

Private communication channel

Each individual knows 2 bits

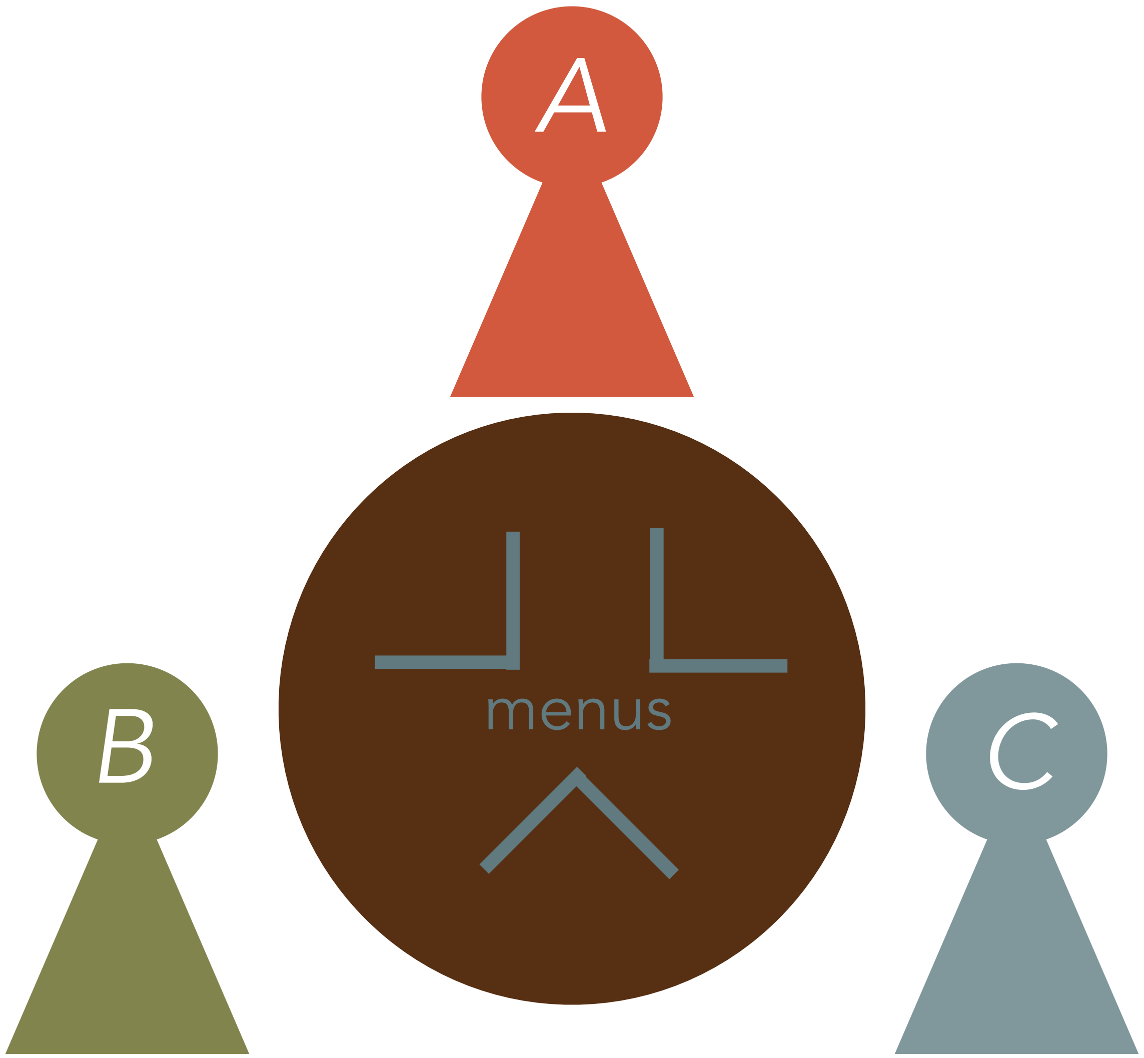
b_{left} and b_{right}

PROBLEM:

One person has a message *m* to send (let's say it's a bit)

Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

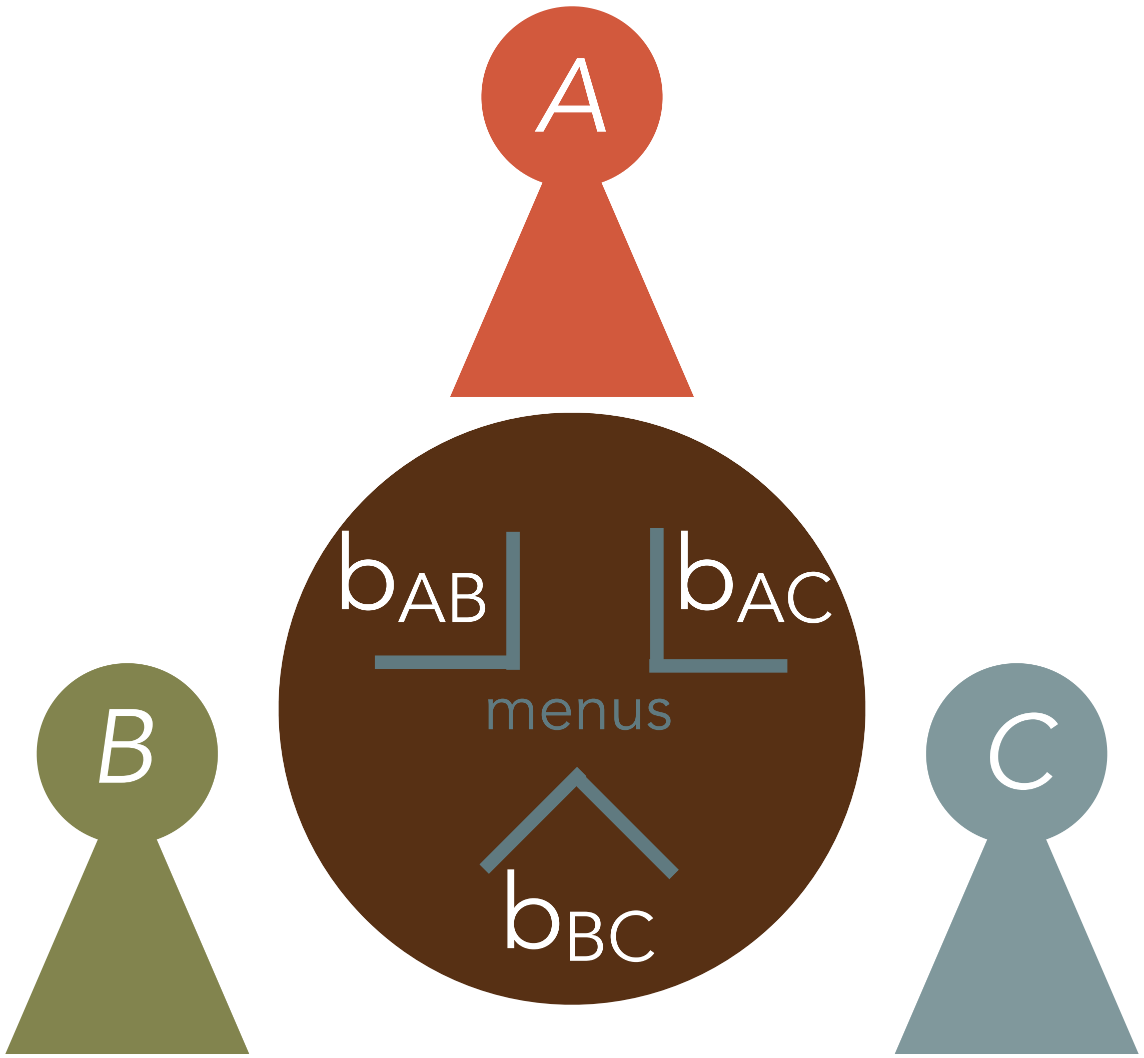
Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

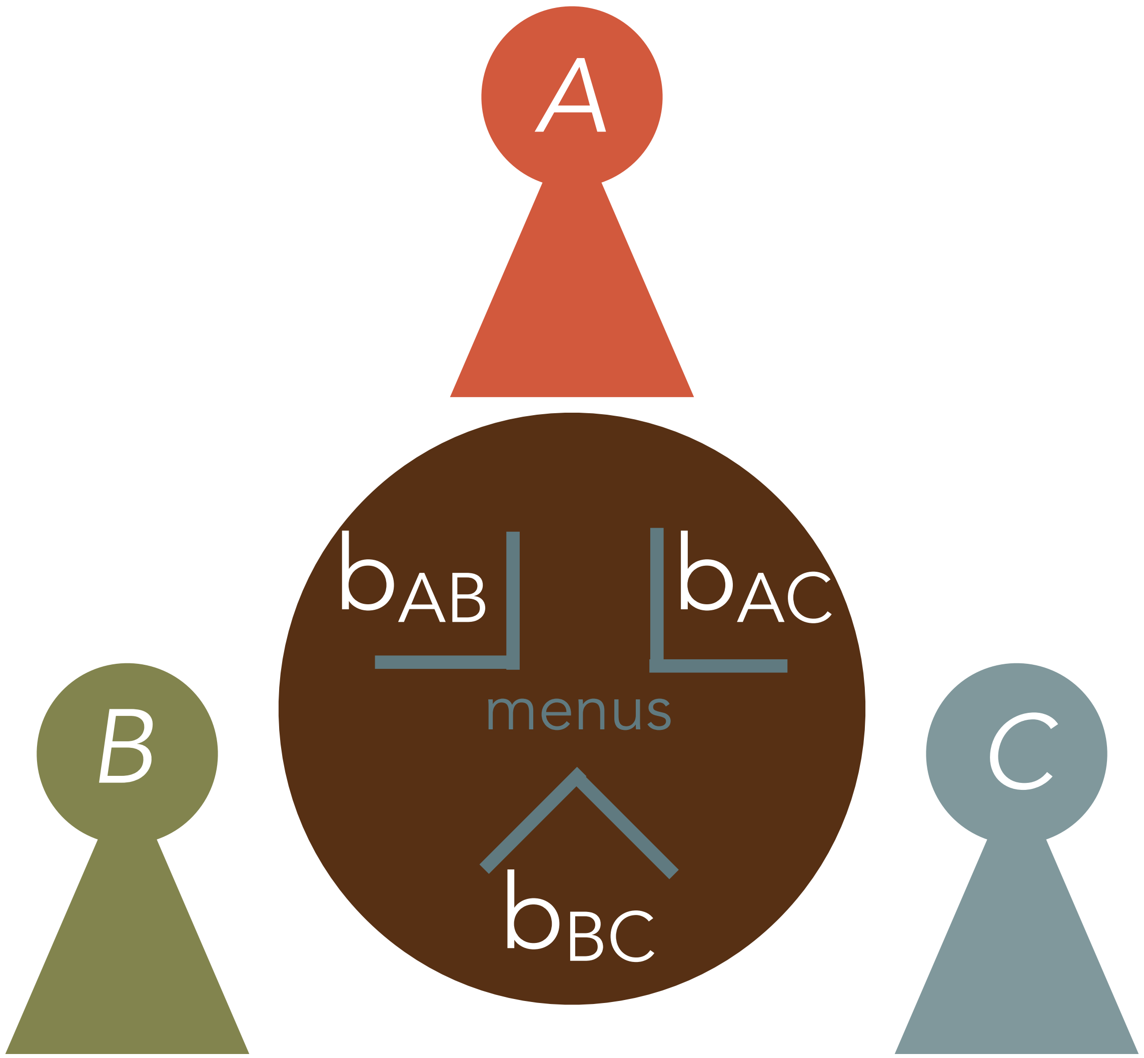
Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

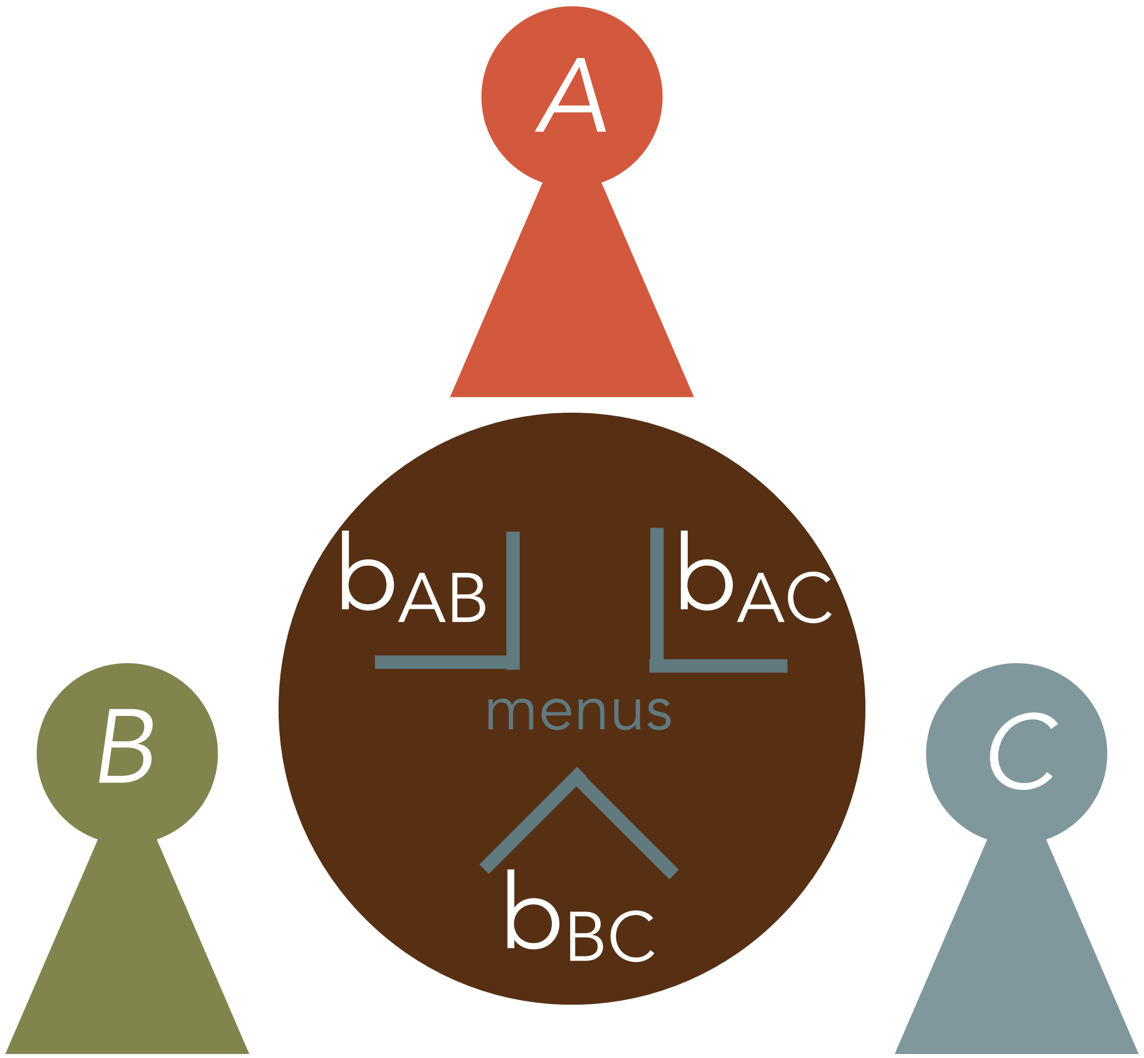
Every individual says a bit
Broadcasts:

If they have *m*:
$m \oplus b_{left} \oplus b_{right}$

Otherwise:
$b_{left} \oplus b_{right}$

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROBLEM:

One person has a message *m* to send (let's say it's a bit)

Can this person reveal that bit without revealing their identity?

PROTOCOL

Each pair props up a menu

Private communication channel

Each pair flips a coin

Gets a shared random bit

Each individual knows 2 bits

b_{left} and b_{right}

Every individual says a bit

Broadcasts:

If they have *m*:

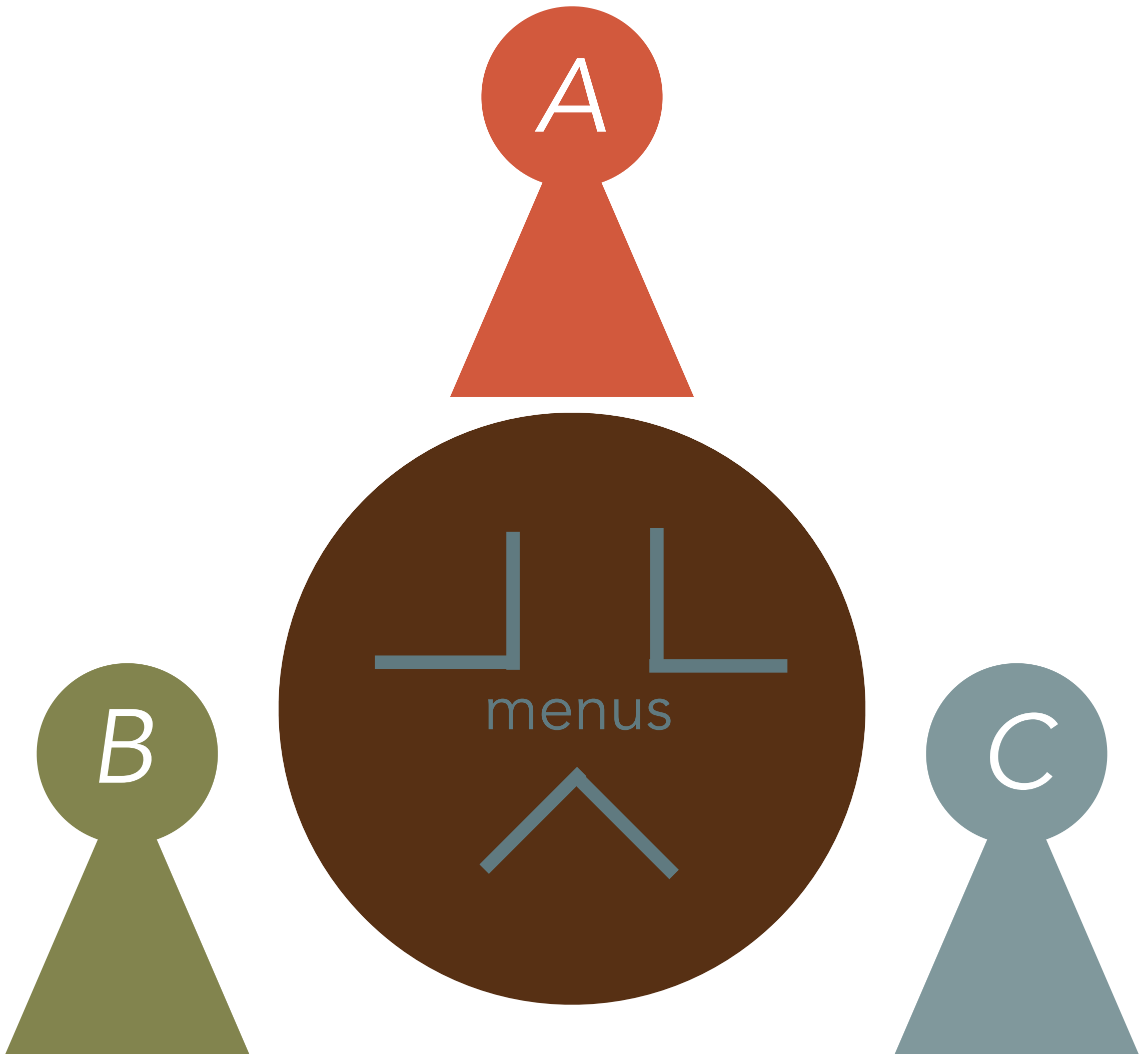
$m \oplus b_{left} \oplus b_{right}$

Otherwise:

$b_{left} \oplus b_{right}$

XOR all messages to recover *m*

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

Every individual says a bit
Broadcasts:

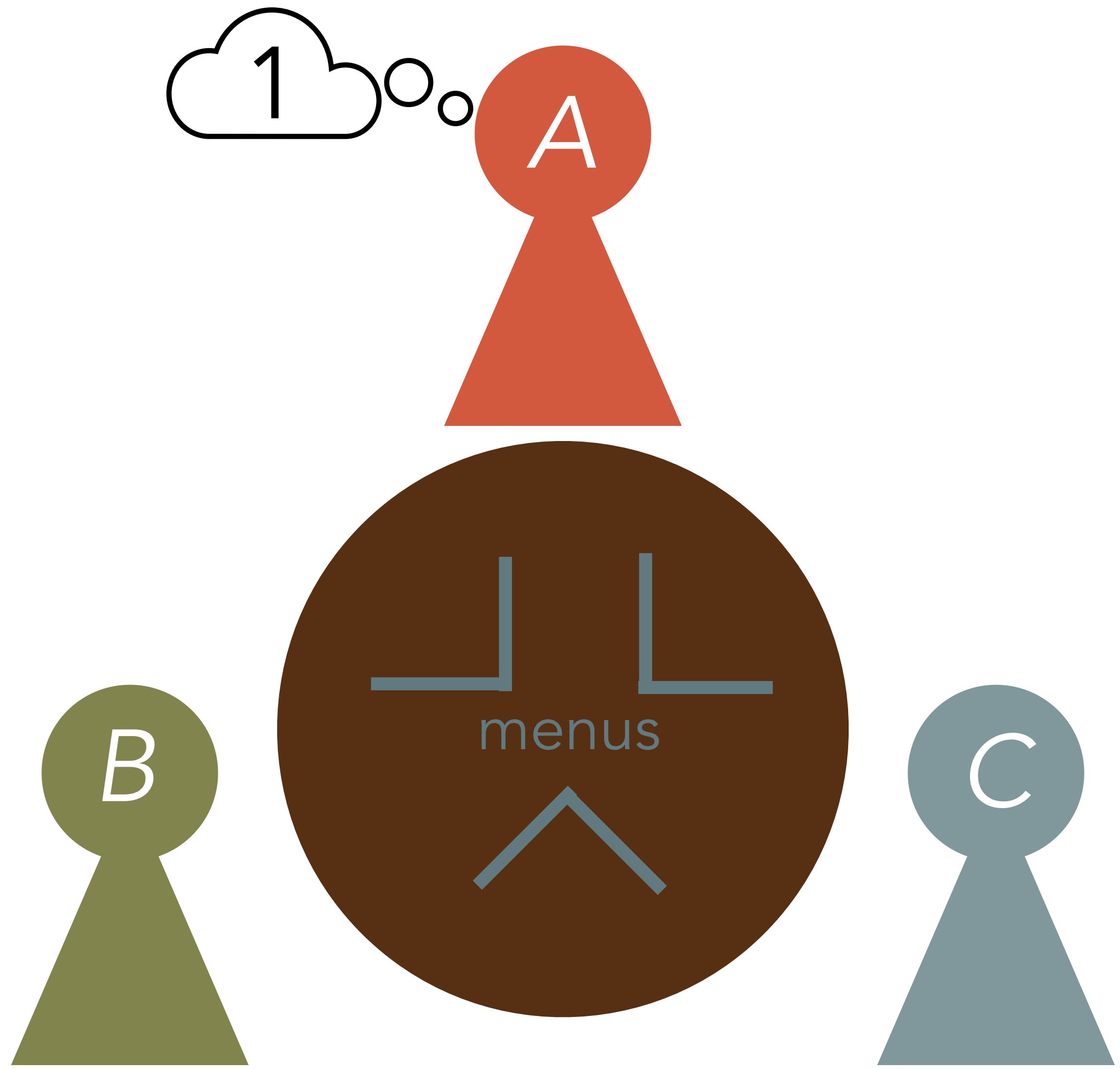
If they have *m*:
$m \oplus b_{left} \oplus b_{right}$

Otherwise:
$b_{left} \oplus b_{right}$

XOR all messages to recover *m*

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

Every individual says a bit
Broadcasts:

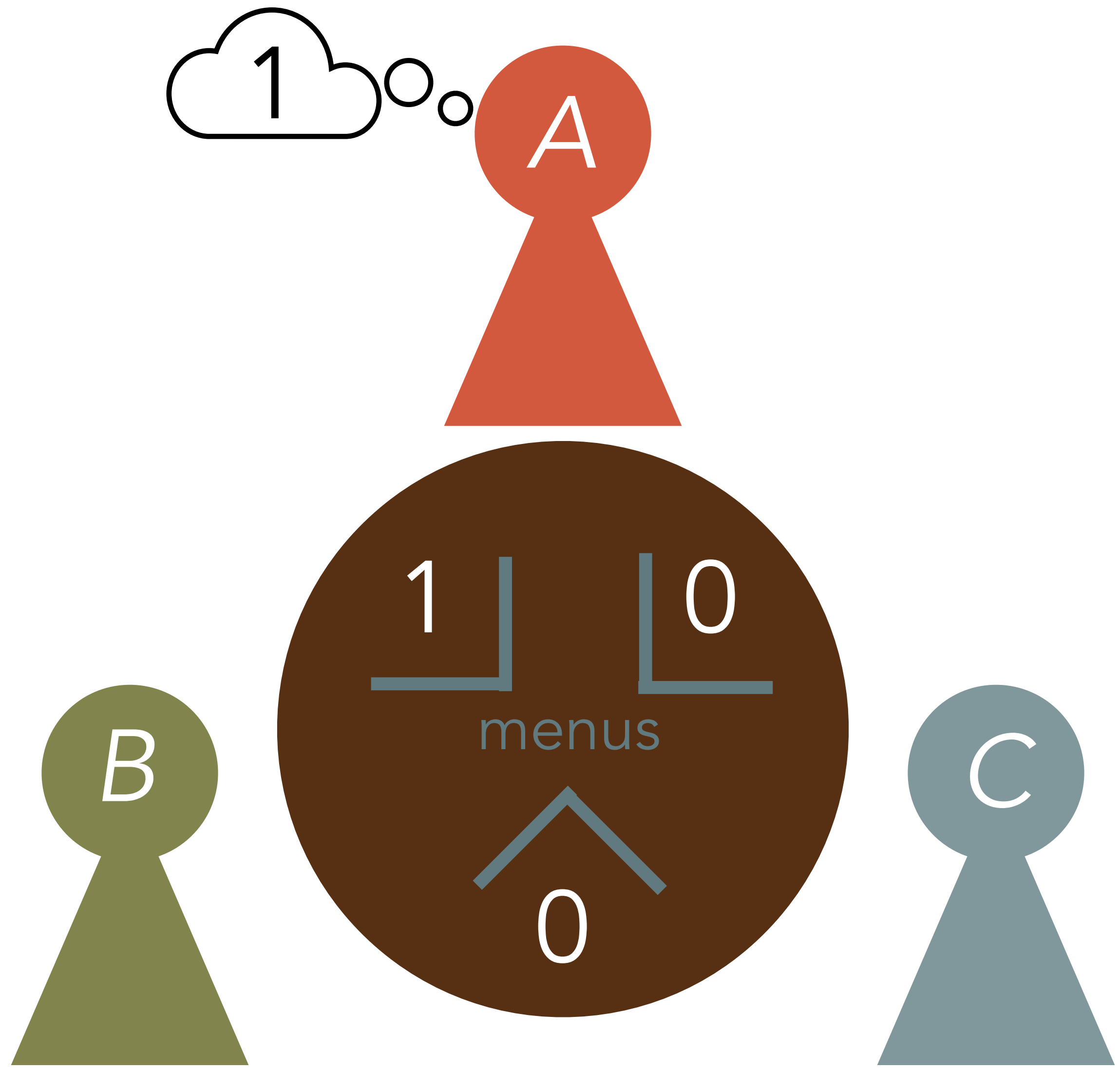
If they have *m*:
$m \oplus b_{left} \oplus b_{right}$

Otherwise:
$b_{left} \oplus b_{right}$

XOR all messages to recover *m*

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

Every individual says a bit
Broadcasts:

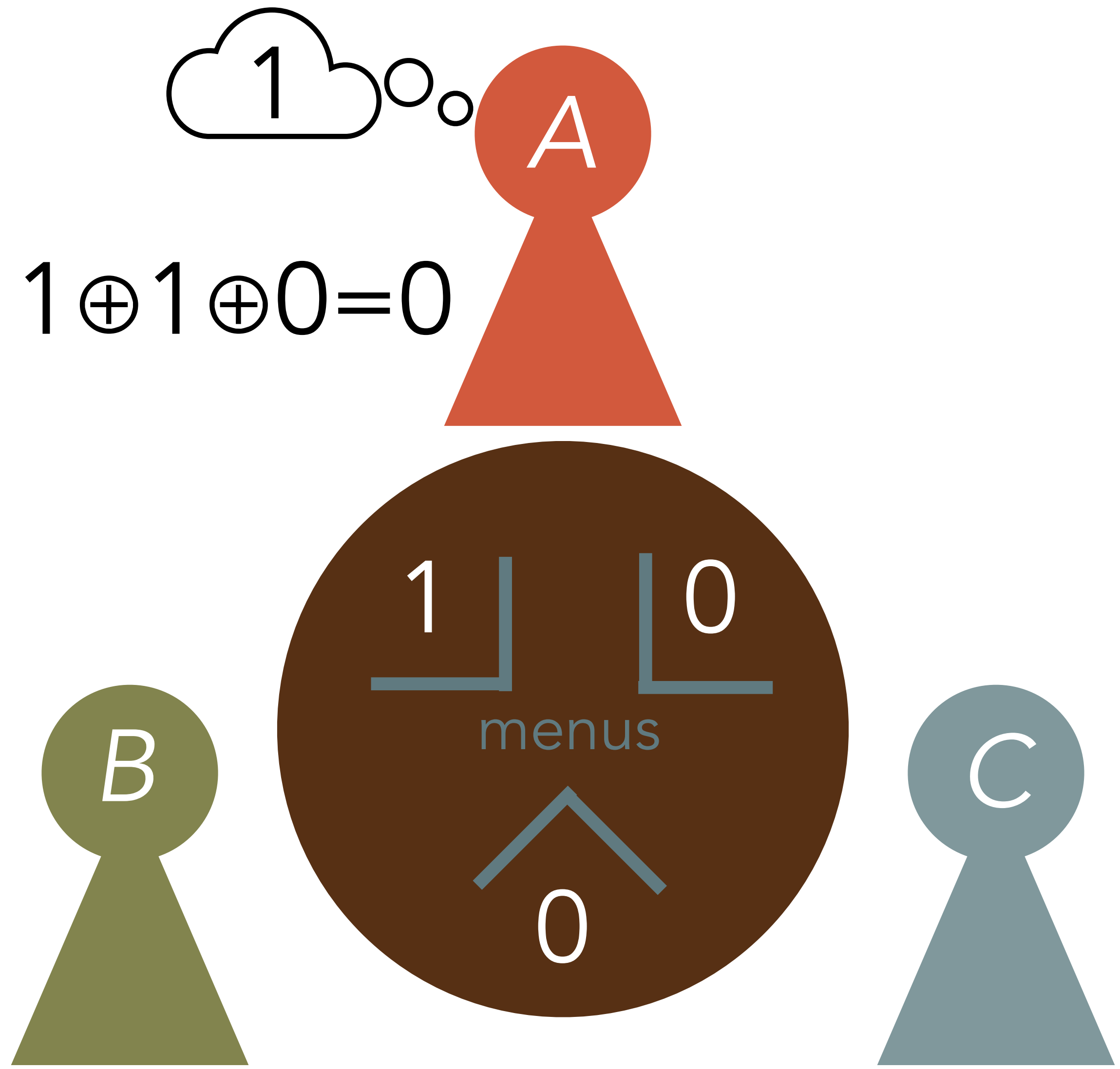
If they have *m*:
$m \oplus b_{left} \oplus b_{right}$

Otherwise:
$b_{left} \oplus b_{right}$

XOR all messages to recover *m*

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

Every individual says a bit
Broadcasts:

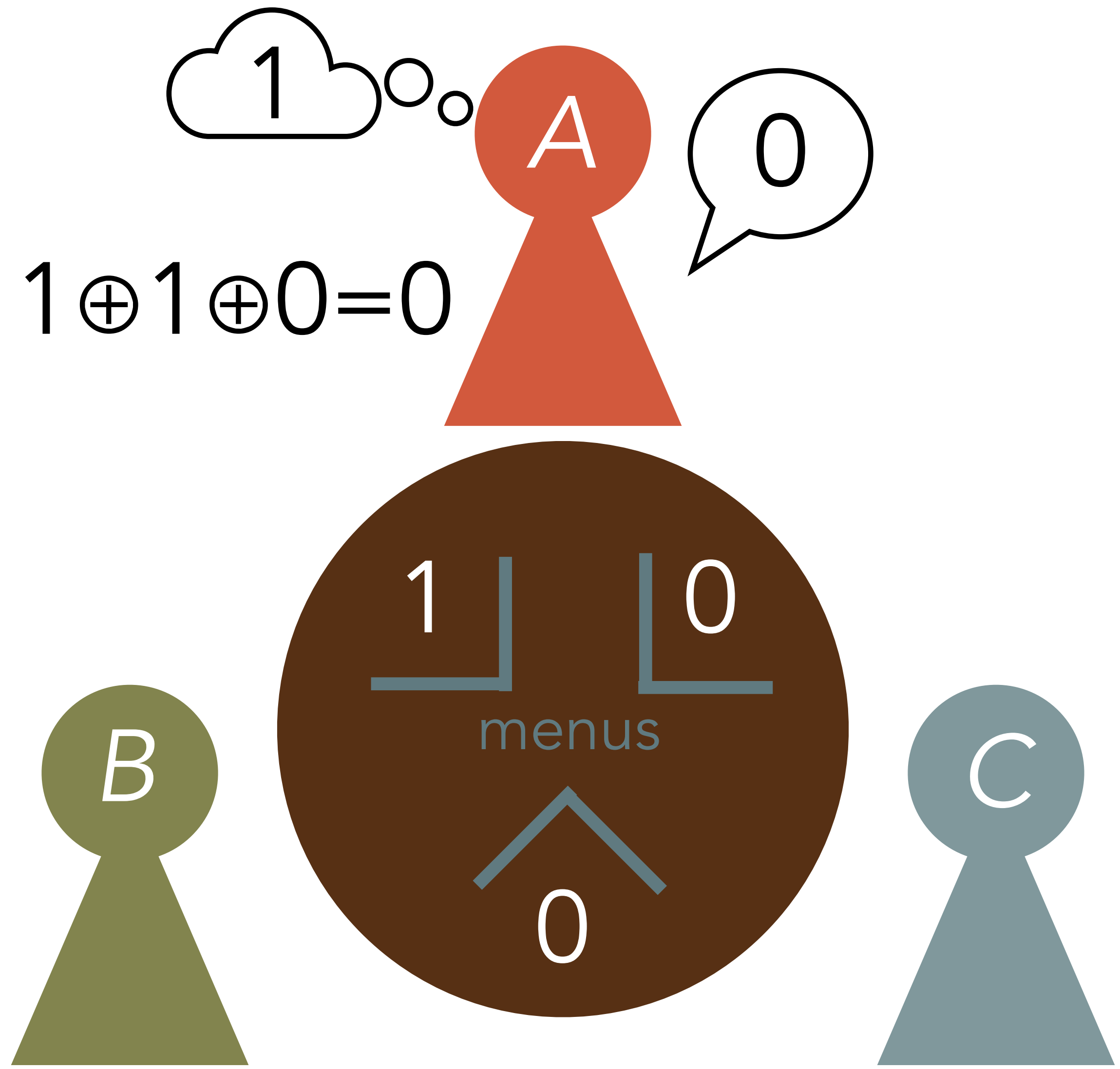
If they have *m*:
$m \oplus b_{left} \oplus b_{right}$

Otherwise:
$b_{left} \oplus b_{right}$

XOR all messages to recover *m*

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

Every individual says a bit
Broadcasts:

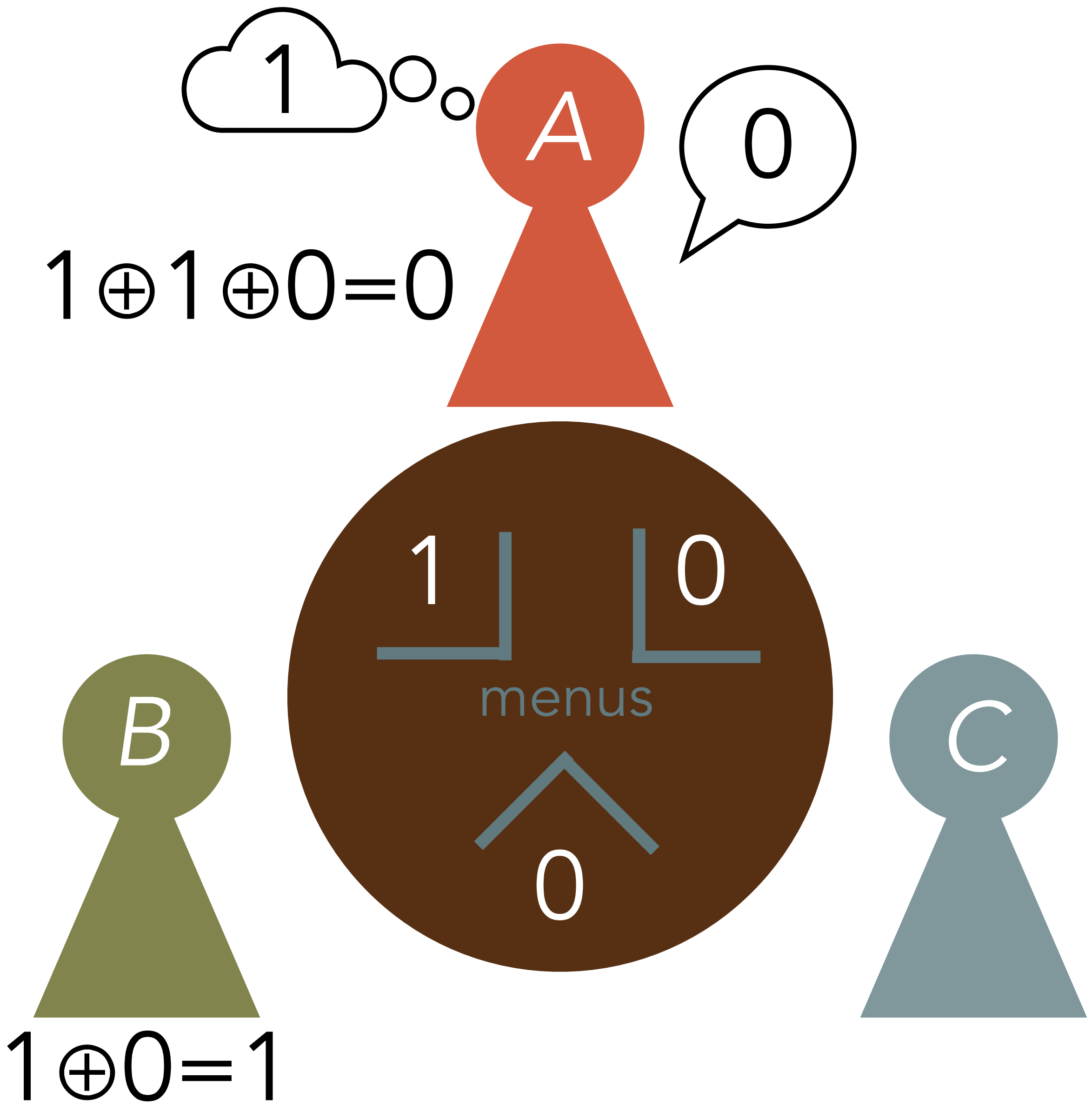
If they have *m*:
$m \oplus b_{left} \oplus b_{right}$

Otherwise:
$b_{left} \oplus b_{right}$

XOR all messages to recover *m*

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu

Private communication channel

Each pair flips a coin

Gets a shared random bit

Each individual knows 2 bits

b_{left} and b_{right}

Every individual says a bit

Broadcasts:

If they have *m*:

$m \oplus b_{left} \oplus b_{right}$

Otherwise:

$b_{left} \oplus b_{right}$

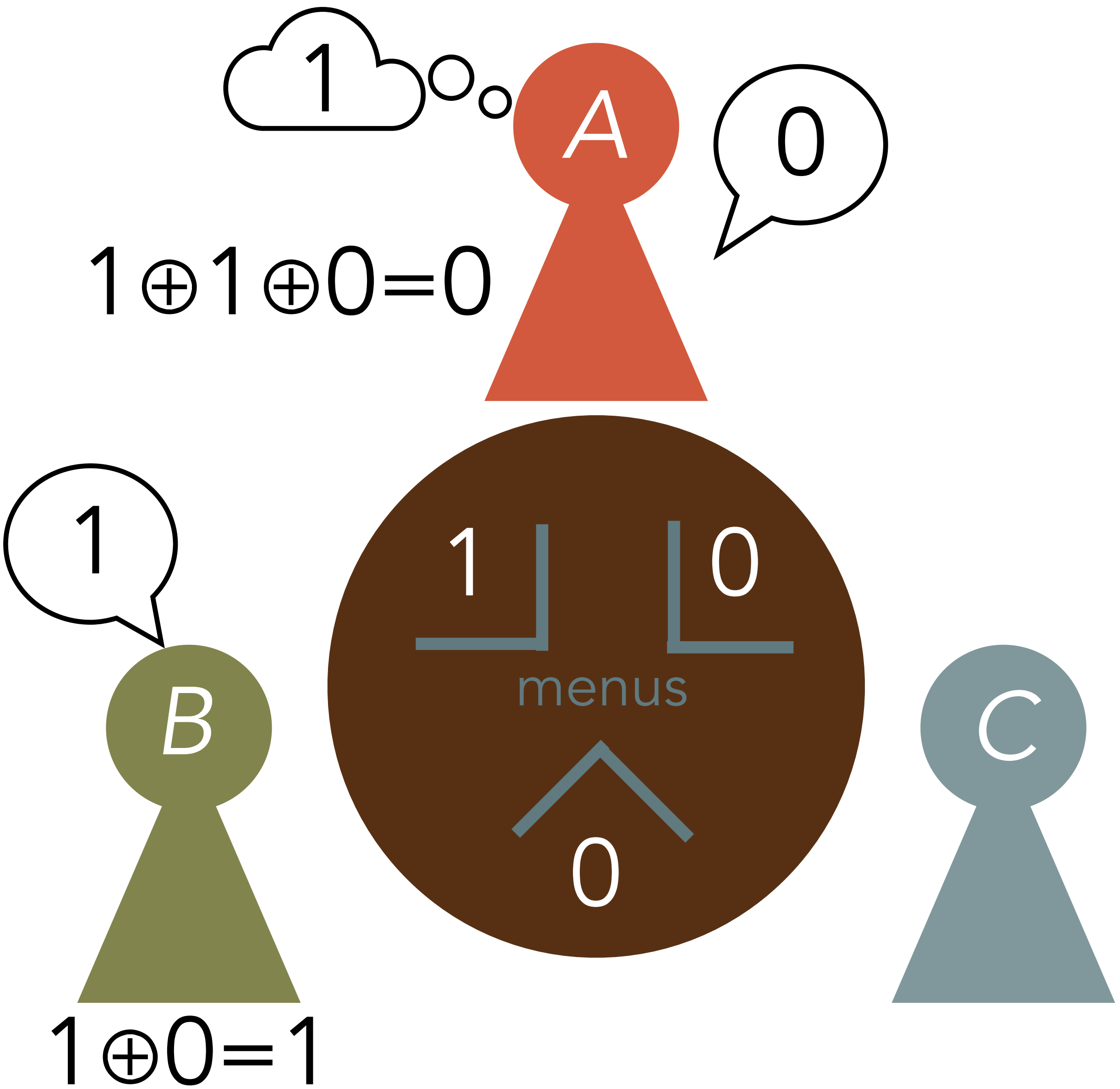
XOR all messages to recover *m*

PROBLEM:

One person has a message *m* to send (let's say it's a bit)

Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROBLEM:

One person has a message *m* to send (let's say it's a bit)

Can this person reveal that bit without revealing their identity?

PROTOCOL

Each pair props up a menu

Private communication channel

Each pair flips a coin

Gets a shared random bit

Each individual knows 2 bits

b_{left} and b_{right}

Every individual says a bit

Broadcasts:

If they have *m*:

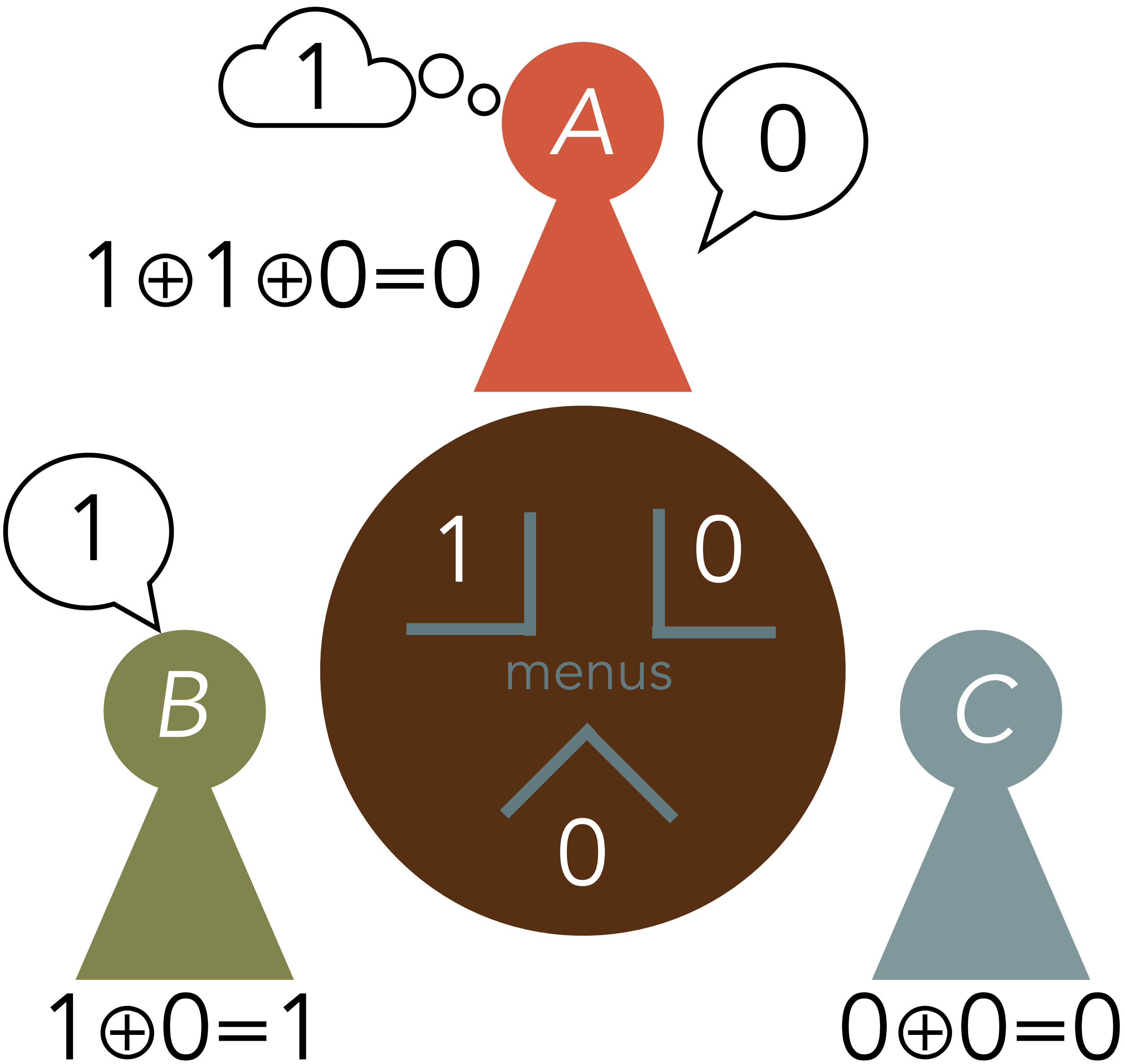
$m \oplus b_{left} \oplus b_{right}$

Otherwise:

$b_{left} \oplus b_{right}$

XOR all messages to recover *m*

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
b_{left} and b_{right}

Every individual says a bit
Broadcasts:

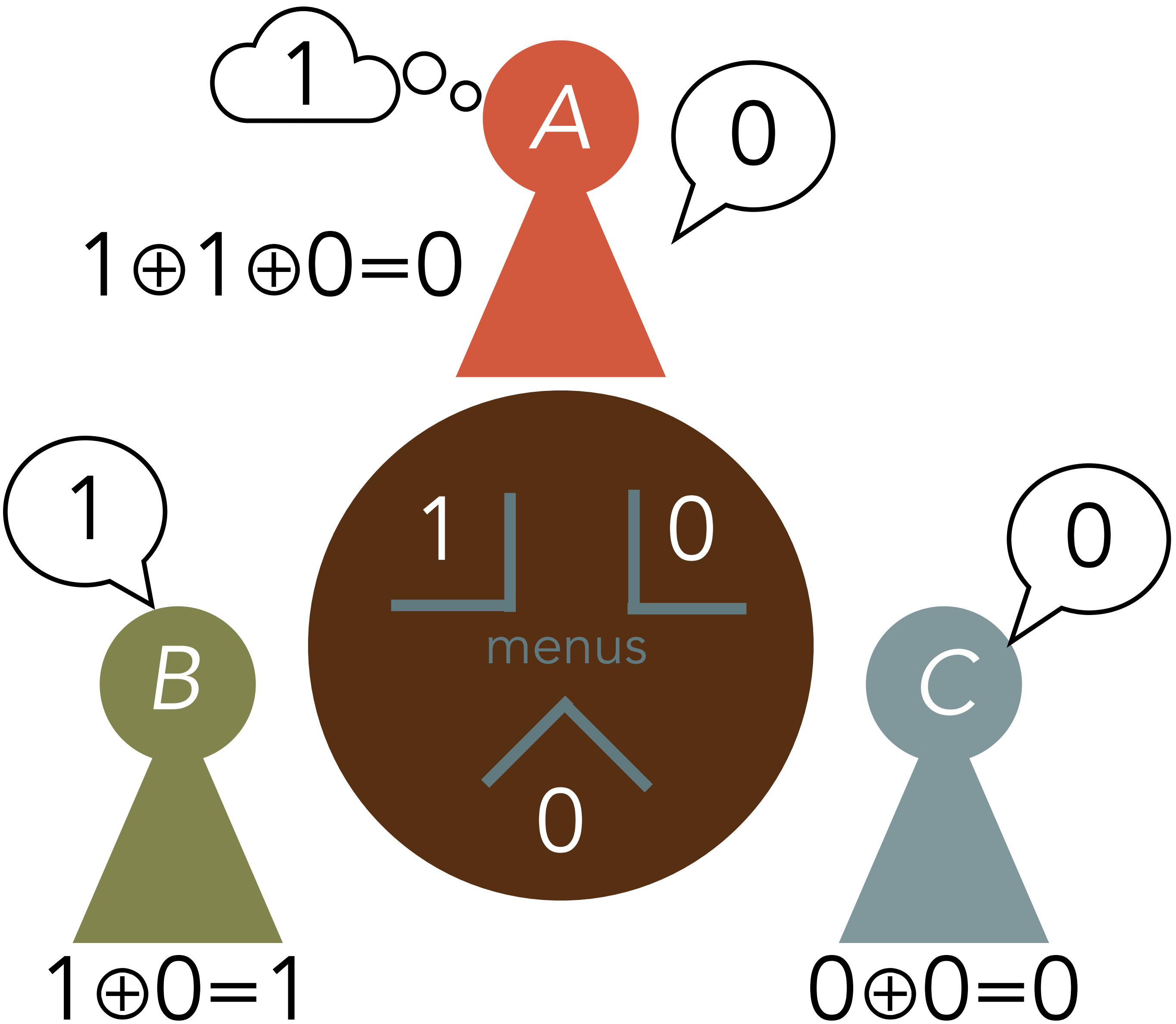
If they have *m*:
$m \oplus b_{left} \oplus b_{right}$

Otherwise:
$b_{left} \oplus b_{right}$

XOR all messages to recover *m*

PROBLEM:
One person has a message *m* to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu

Private communication channel

Each pair flips a coin

Gets a shared random bit

Each individual knows 2 bits

b_{left} and b_{right}

Every individual says a bit

Broadcasts:

If they have *m*:

$m \oplus b_{left} \oplus b_{right}$

Otherwise:

$b_{left} \oplus b_{right}$

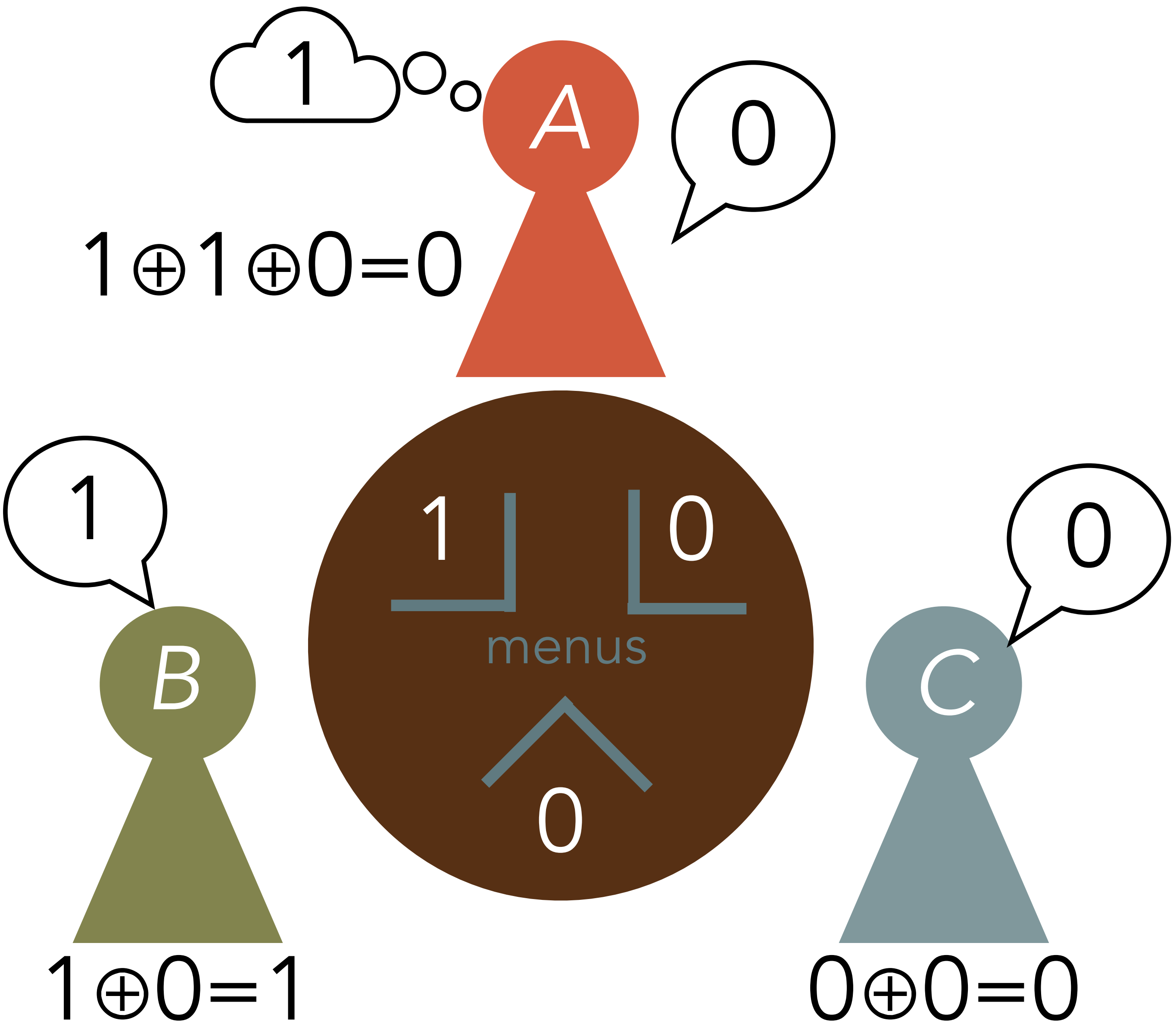
XOR all messages to recover *m*

PROBLEM:

One person has a message *m* to send (let's say it's a bit)

Can this person reveal that bit without revealing their identity?

THE DINING CRYPTOGRAPHER'S PROBLEM



PROTOCOL

Each pair props up a menu

Private communication channel

Each pair flips a coin

Gets a shared random bit

Each individual knows 2 bits

b_{left} and b_{right}

Every individual says a bit

Broadcasts:

If they have *m*:

$m \oplus b_{left} \oplus b_{right}$

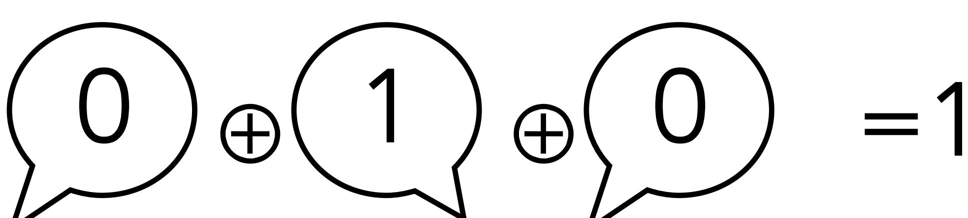
Otherwise:

$b_{left} \oplus b_{right}$

PROBLEM:

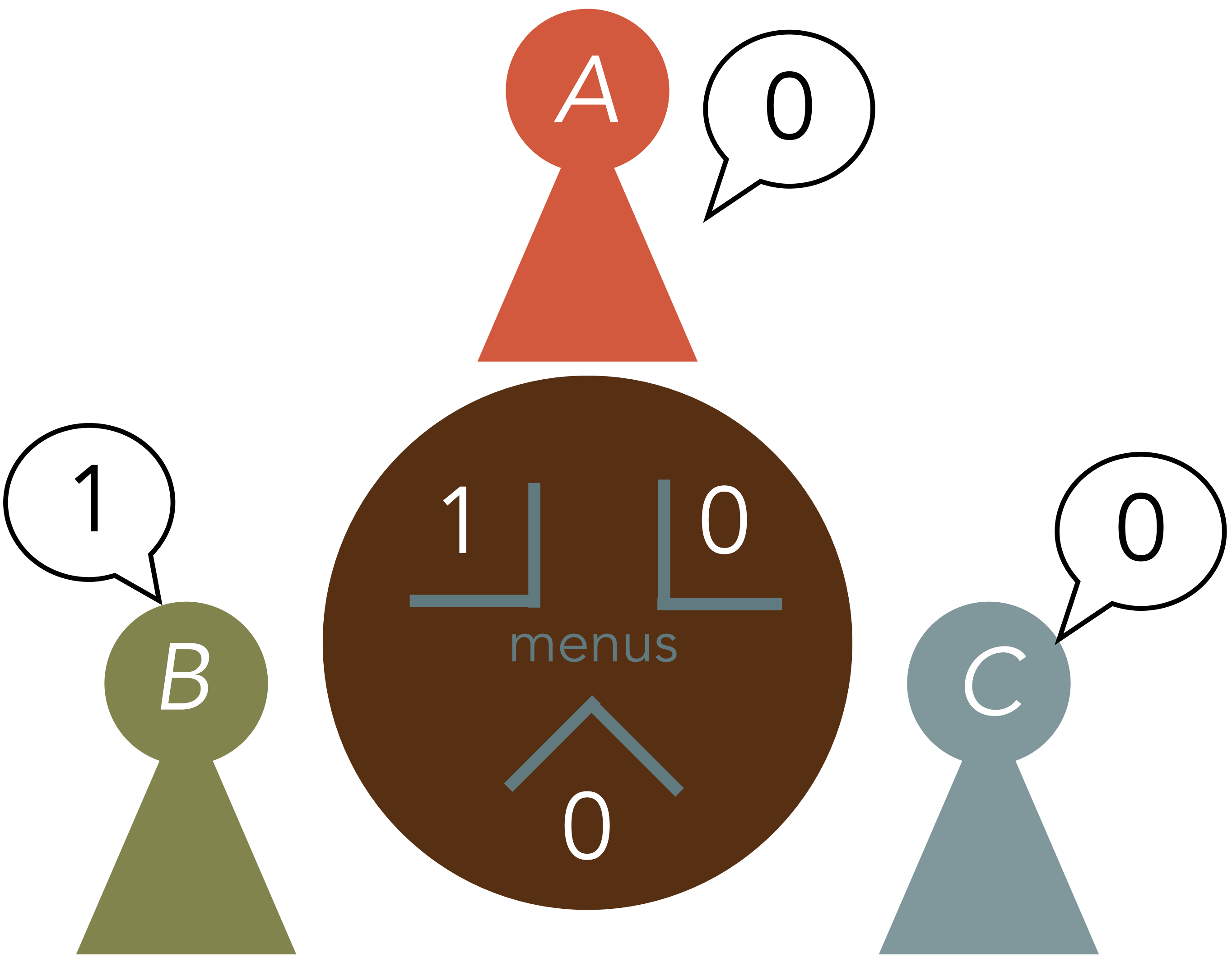
One person has a message *m* to send (let's say it's a bit)

Can this person reveal that bit without revealing their identity?



XOR all messages to recover *m*

WHO LEARNS WHAT?



AFTER THE PROTOCOL

Everyone knows

THEIR b_{left} and b_{right}

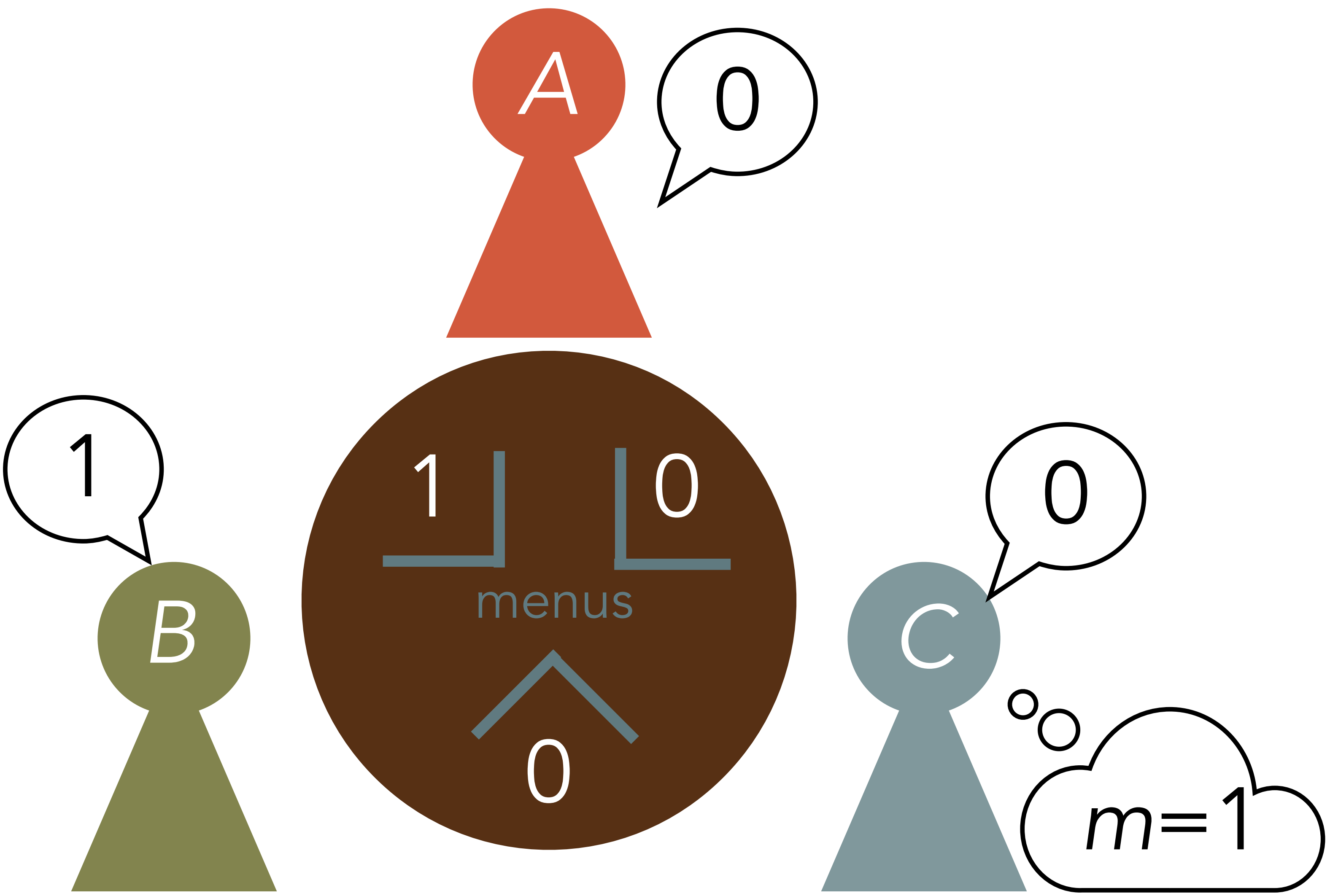
The message *m*

(Whether or not they sent it)

No one learns

The remaining bit

WHO LEARNS WHAT?



AFTER THE PROTOCOL

Everyone knows

THEIR b_{left} and b_{right}

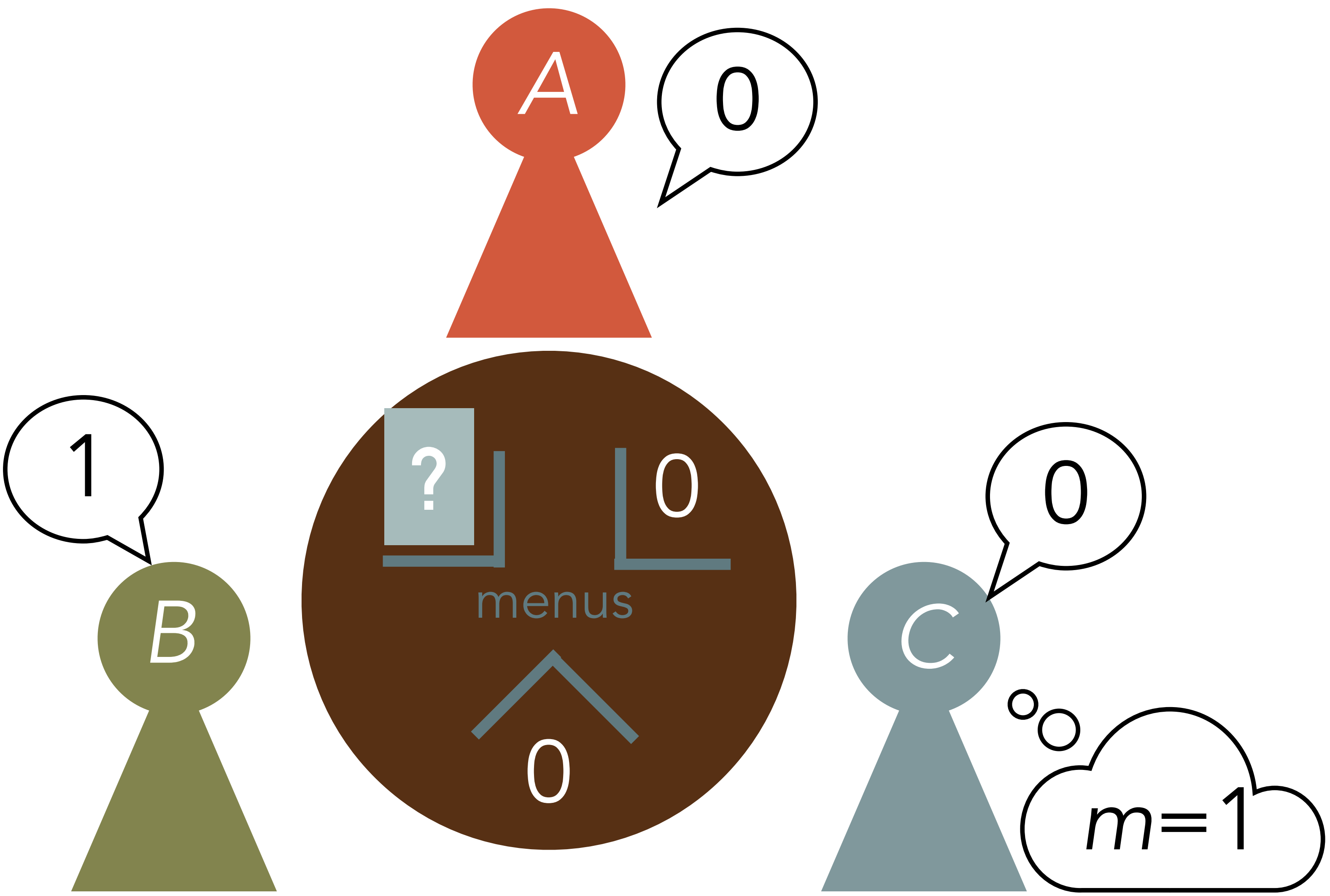
The message *m*

(Whether or not they sent it)

No one learns

The remaining bit

WHO LEARNS WHAT?



AFTER THE PROTOCOL

Everyone knows

THEIR b_{left} and b_{right}

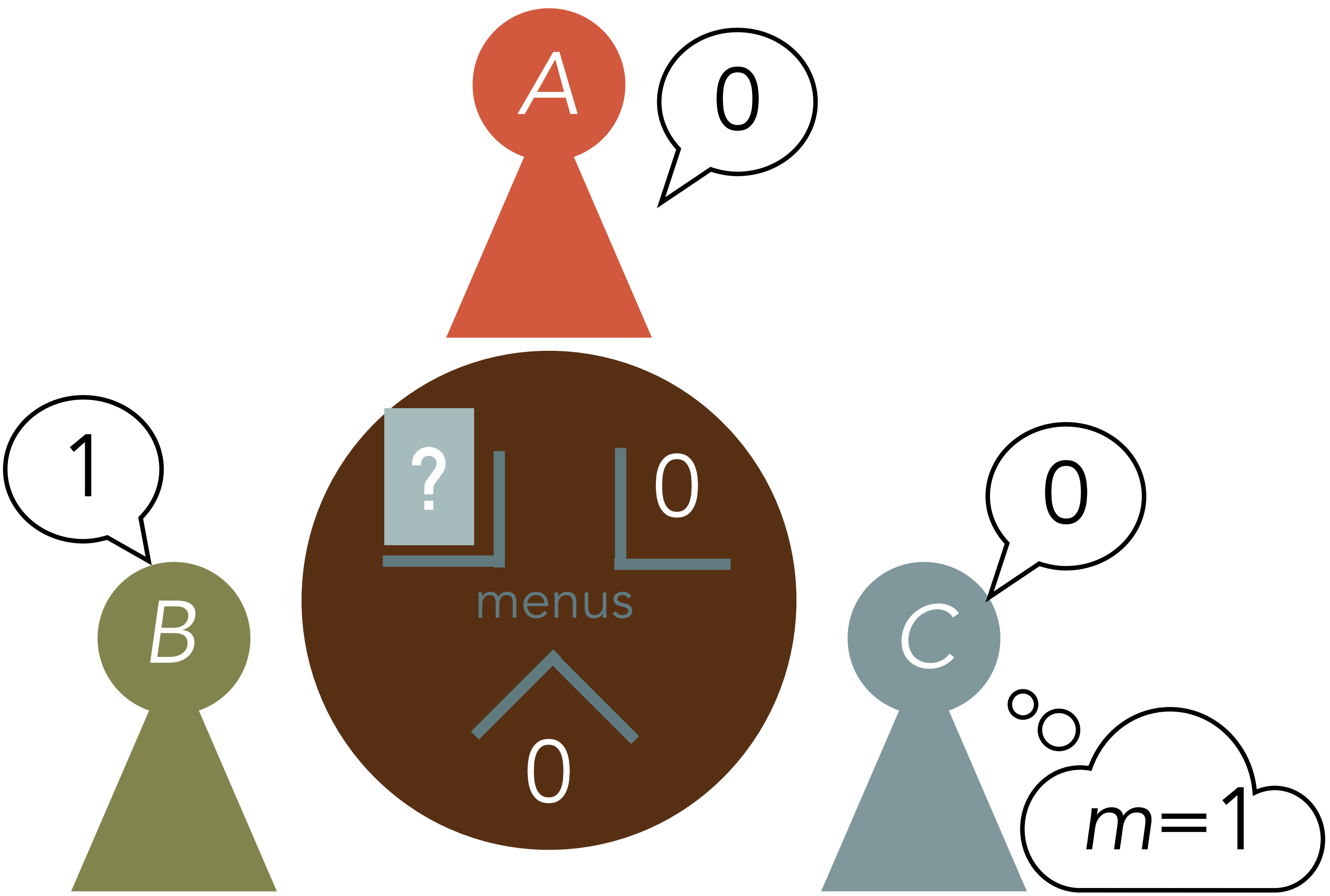
The message *m*

(Whether or not they sent it)

No one learns

The remaining bit

WHO LEARNS WHAT?



AFTER THE PROTOCOL

Everyone knows

THEIR b_{left} and b_{right}

The message *m*

(Whether or not they sent it)

No one learns

The remaining bit

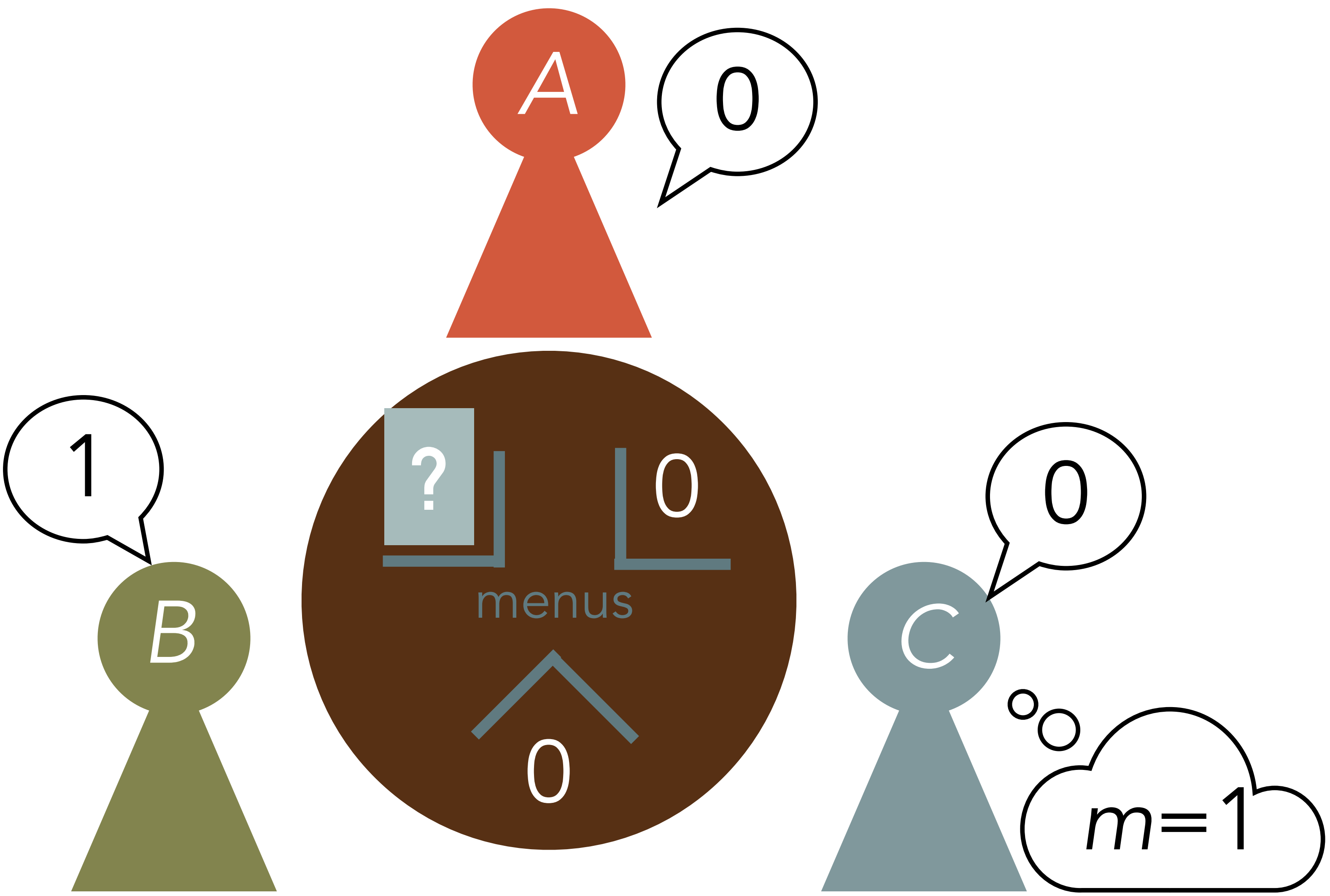
If b_{AB} = 0

If A sent *m*, he would have sent $0 \oplus 0 + 1 = 1$ ✗

If B sent *m*, he would have sent $1 \oplus 0 \oplus 0 = 1$ ✓

Therefore, if $b_{AB} = 0$ then B was the sender

WHO LEARNS WHAT?



AFTER THE PROTOCOL

Everyone knows

THEIR b_{left} and b_{right}

The message *m*

(Whether or not they sent it)

No one learns

The remaining bit

If b_{AB} = 0

If A sent *m*, he would have sent $0 \oplus 0 + 1 = 1$ ✗

If B sent *m*, he would have sent $1 \oplus 0 \oplus 0 = 1$ ✓

Therefore, if $b_{AB} = 0$ then B was the sender

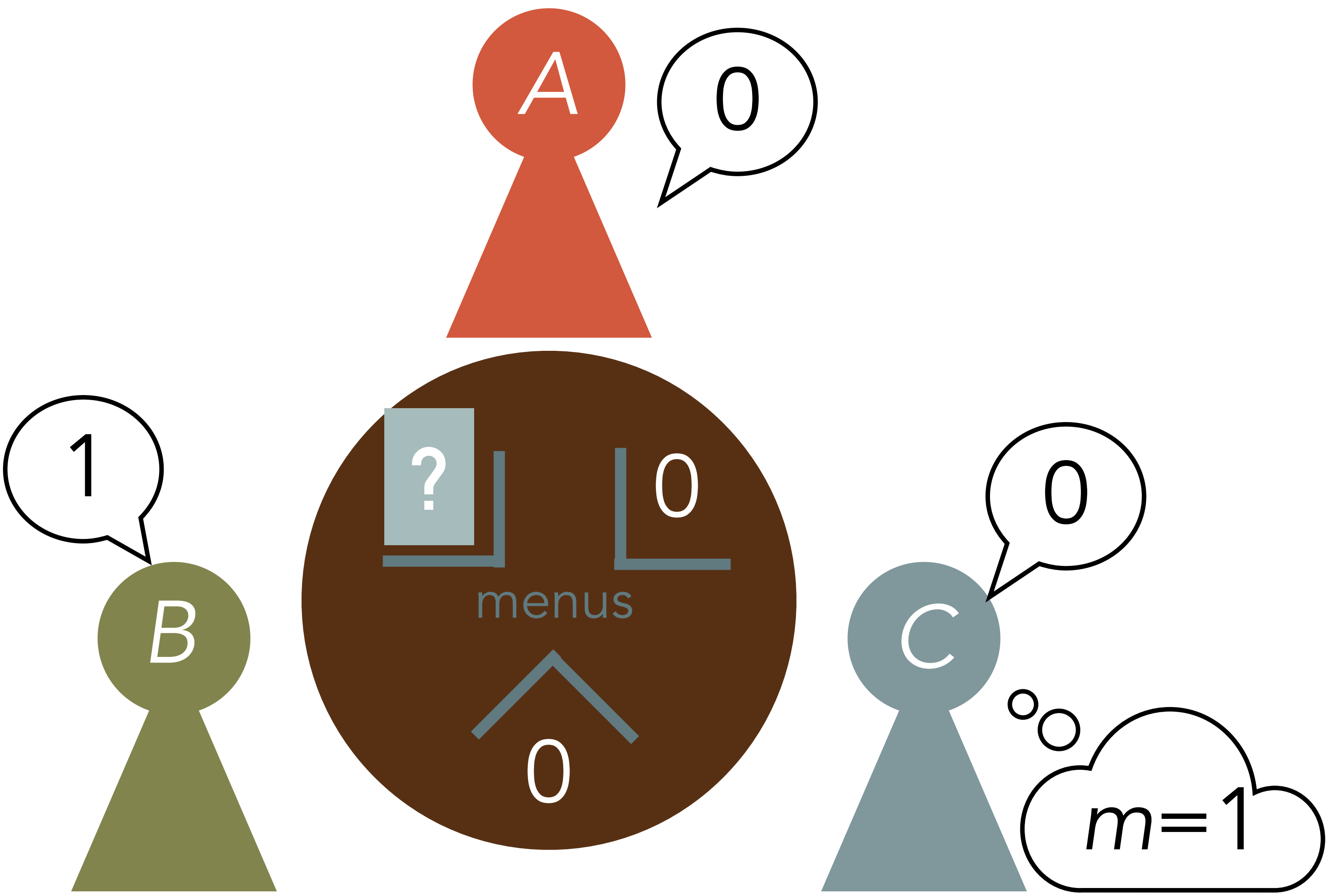
If b_{AB} = 1

If A sent *m*, he would have sent $1 \oplus 1 \oplus 0 = 0$ ✓

If B sent *m*, he would have sent $1 \oplus 0 \oplus 0 = 1$ ✗

Therefore, if $b_{AB} = 1$ then A was the sender

WHO LEARNS WHAT?



AFTER THE PROTOCOL

Everyone knows

THEIR b_{left} and b_{right}

The message *m*

(Whether or not they sent it)

No one learns

The remaining bit

If b_{AB} = 0

If A sent *m*, he would have sent 0⊕0+1 = 1 ✗

If B sent *m*, he would have sent 1⊕0⊕0 = 1 ✓

Therefore, if b_{AB} = 0 then B was the sender

If b_{AB} = 1

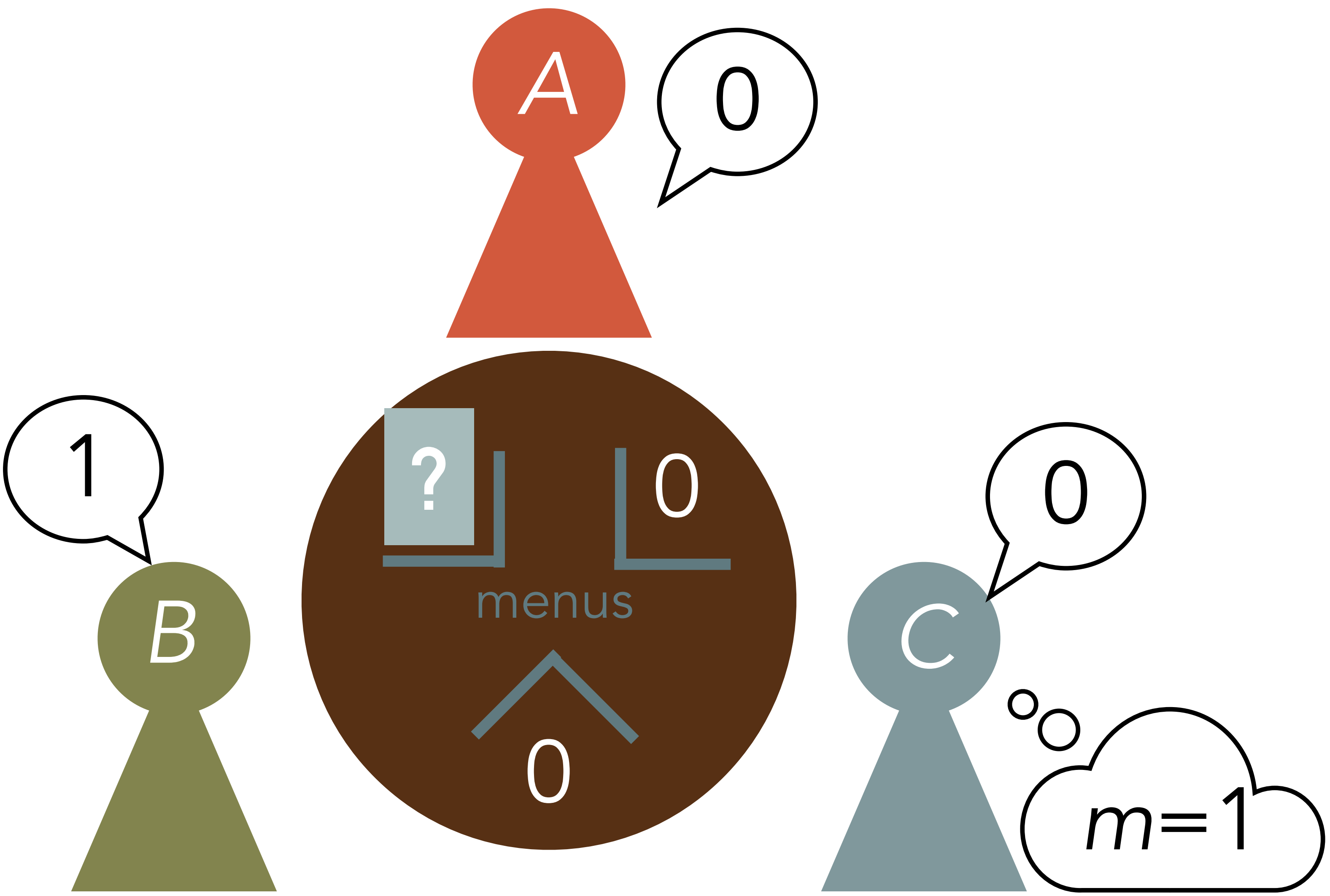
If A sent *m*, he would have sent 1⊕1⊕0 = 0 ✓

If B sent *m*, he would have sent 1⊕0⊕0 = 1 ✗

Therefore, if b_{AB} = 1 then A was the sender

Each of these has probability 50% (it was determined by a coin flip)

WHO LEARNS WHAT?



AFTER THE PROTOCOL

Everyone knows

THEIR b_{left} and b_{right}

The message *m*

(Whether or not they sent it)

No one learns

The remaining bit

No one learns

Any information about who sent the message

If b_{AB} = 0

If A sent *m*, he would have sent 0⊕0+1 = 1 ✗

If B sent *m*, he would have sent 1⊕0⊕0 = 1 ✓

Therefore, if b_{AB} = 0 then B was the sender

If b_{AB} = 1

If A sent *m*, he would have sent 1⊕1⊕0 = 0 ✓

If B sent *m*, he would have sent 1⊕0⊕0 = 1 ✗

Therefore, if b_{AB} = 1 then A was the sender

Each of these has probability 50% (it was determined by a coin flip)

DINING CRYPTOGRAPHERS IN PRACTICE

INSTEAD OF SENDING BITS

Send streams of packets; flip multiple coints

HOW CAN MORE THAN ONE PERSON NEEDS TO SEND A MESSAGE?

Take turns? But what happens when two try to send at once?

DIFFICULT BUT NOT IMPOSSIBLE TO SCALE UP

In practice we use something else…

RELATED PAPERS

J. Cryptology (1988) 1: 65–75

Journal of Cryptology

The Dining Cryptographers Problem: *Unconditional Sender and Recipient Untraceability*

David Chaum

Centre for Mathematics and Computer Science, Kruislan 413, 1098 SJ Amsterdam, The Netherlands

Abstract. Keeping confidential who sends which messages, in a world where any physical transmission can be traced to its origin, seems impossible. The solution presented here is unconditionally or cryptographically secure, depending on whether it is based on one-time-use keys or on public keys, respectively. It can be adapted to address efficiently a wide variety of practical considerations.

Key words. Untraceability, Unconditional Security, Pseudonymity.

Introduction

Three cryptographers are sitting down to dinner at their favorite three-star restaurant. Their waiter informs them that arrangements have been made with the maître d'hôtel for the bill to be paid anonymously. One of the cryptographers might be paying for the dinner, or it might have been NSA (U.S. National Security Agency). The three cryptographers respect each other's right to make an anonymous payment, but they wonder if NSA is paying. They resolve their uncertainty fairly by carrying out the following protocol:

Each cryptographer flips an unbiased coin behind his menu, between him and the cryptographer on his right, so that only the two of them can see the outcome. Each cryptographer then states aloud whether the two coins he can see—the one he flipped and the one his left-hand neighbor flipped—fell on the same side or on different sides. If one of the cryptographers is the payer, he states the opposite of what he sees. An odd number of differences uttered at the table indicates that a cryptographer is paying; an even number indicates that NSA is paying (assuming that the dinner was paid for only once). Yet if a cryptographer is paying, neither of the other two learns anything from the utterances about which cryptographer it is.

To see why the protocol is unconditionally secure if carried out faithfully, consider the dilemma of a cryptographer who is not the payer and wishes to find out which cryptographer is. (If NSA pays, there is no anonymity problem.) There are two cases. In case (1) the two coins he sees are the same, one of the other cryptographers said "different," and the other one said "same." If the hidden outcome was the same as the two outcomes he sees, the cryptographer who said "different" is the payer; if the outcome was different, the one who said "same" is the payer. But since the hidden coin is fair, both possibilities are equally likely. In case (2) the coins he sees are

---

Technical Note
Programming Techniques and Data Structures

R. Rivest
Editor

Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms

David L. Chaum
University of California, Berkeley

A technique based on public key cryptography is presented that allows an electronic mail system to hide who a participant communicates with as well as the content of the communication—in spite of an unsecured underlying telecommunication system. The technique does not require a universally trusted authority. One correspondent can remain anonymous to a second, while allowing the second to respond via an untraceble return address.

The technique can also be used to form rosters of untraceable digital pseudonyms from selected applications. Applicants retain the exclusive ability to form digital signatures corresponding to their pseudonyms. Elections in which any interested party can verify that the ballots have been properly counted are possible if anonymously mailed ballots are signed with pseudonyms from a roster of registered voters. Another use allows an individual to correspond with a record-keeping organization under a unique pseudonym which appears in a roster of acceptable clients.

Key Words and Phrases: electronic mail, public key cryptosystems, digital signatures, traffic analysis, security, privacy

CR Categories: 2.12, 3.81

Introduction

Cryptology is the science of secret communication. Cryptographic techniques have been providing secrecy of message content for thousands of years [3]. Recently, some new solutions to the "key distribution problem" (the problem of providing each communicant with a secret key) have been suggested [2, 4], under the name of public key cryptography. Another cryptographic problem, "the traffic analysis problem" (the problem of keeping confidential who converses with whom, and when they converse), will become increasingly important with the growth of electronic mail. This paper presents a solution to the traffic analysis problem that is based on public key cryptography. Baran has solved the traffic analysis problem for networks [1], but requires each participant to trust a common authority. In contrast, systems based on the solution advanced here can be compromised only by subversion or conspiracy of all of a set of authorities. Ideally, each participant is an authority.

The following two sections introduce the notation and assumptions. Then the basic concepts are introduced for some special cases involving a series of one or more authorities. The final section covers general purpose mail networks.

This work was partially supported by the National Science Foundation under Grant MCS 75-23739 and by the Air Force Office of Scientific Research under Contract F49620-79-CO173.

Author's present address: Computer Science Division, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, California 94720. (415) 642-1024.

Notation

Someone becomes a user of a public key cryptosystem (like that of Rivest, Shamir, and Adleman [5]) by creating a pair of keys K and K^{-1} from a suitable randomly generated seed. The public key K is made known to the other users or anyone else who cares to know it; the private key K^{-1} is never divulged. The encryption of X with key K will be denoted $K(X)$, and is just the image of X under the mapping implemented by the cryptographic algorithm using key K. The increased utility of these algorithms over conventional algorithms results because the two keys are inverses of each other, in the sense that

$$K^{-1}(K(X)) = K(K^{-1}(X)) = X.$$

A message X is *sealed* with a public key K so that only the holder of the private key K^{-1} can discover its content. If X is simply encrypted with K, then anyone could verify a guess that $Y = X$ by checking whether $K(Y) = K(X)$. This threat can be eliminated by attaching a large string of random bits R to X before encrypting. The sealing of X with K is then denoted $K(R, X)$. A user *signs* some material X by prepending a large constant C (all zeros, for example) and then encrypting with its private key, denoted $K^{-1}(C, X) = Y$. Anyone can verify that Y has been signed by the holder of K^{-1} and determine the signed matter X, by forming $K(Y) = C, X$, and checking for C.

Assumptions

The approach taken here is based on two important assumptions: